Access Manager Appliance 4.2
Administration Guide

November 2015

NetIQ

Legal Notice

For information about NetIQ legal notices, disclaimers, warranties, export and other use restrictions, U.S. Government restricted rights, patent policy, and FIPS compliance, see https://www.netiq.com/company/legal/.

© 2015 NetIQ Corporation. All Rights Reserved.

For information about NetIQ trademarks, see https://www.netiq.com/company/legal/. All third-party trademarks are the property of their respective owners.

Contents

About NetIQ Corporation 17
About this Book and the Library 19

1 Overview 21

How Access Manager Appliance Solves Business Challenges . . . 21
Protecting Resources While Providing Access . . . 22
Managing Passwords with Single Sign-On . . . 23
Enforcing Business Policies . . . 24
Sharing Identity Information . . . 25
Protecting Identity Information . . . 27
Complying with Regulations . . . 28
How Access Manager Appliance Works . . . 28
Authentication . . . 29
Authorization . . . 29
Identity Injection . . . 29
Identity Federation . . . 29
Access Manager Appliance Components and Their Features . . . 30
Administration Console . . . 30
Identity Servers . . . 31
Access Gateways . . . 32
MobileAccess . . . 33
User Portal . . . 33
Language Support . . . 33

Part I Configuring Access Manager 35

2 Configuring the Default View 37

Configuring the Default View . . . 37
Changing the View . . . 39
Setting a Permanent Default View . . . 39
Managing the Administration Console Session Timeout . . . 39
Managing Administrators . . . 39
Creating Multiple Admin Accounts . . . 40
Managing Policy View Administrators . . . 41
Managing Delegated Administrators . . . 41
Changing Administrator's Password . . . 46
Changing the IP Address of Access Manager Appliance . . . 47
Changing the DNS Name of Access Manager Appliance . . . 48

3 Setting Up a Basic Access Manager Appliance Configuration 49

Understanding Access Manager Appliance Process Flow . . . 50
Prerequisites for Setup . . . 51
Setting up User Stores for Identity Server Configuration . . . 52
Identity Servers Cluster . . . 52
Managing a Cluster of the Identity Servers . . . 52
Configuring the Identity Server Shared Settings . . . 60
Configuring Attribute Sets . . . 61

Editing Attribute Sets . . . 63
Configuring User Matching Expressions . . . 63
Adding Custom Attributes . . . 64
Adding Authentication Card Images . . . 66
Creating an Image Set . . . 67
Metadata Repositories . . . 67
User Attribute Retrieval and Transformation . . . 68
Configuring the Access Gateway . . . 85
Configuring a Reverse Proxy . . . 85
Configuring a Public Protected Resource . . . 87
Setting Up Policies . . . 88
Access Gateways Clusters . . . 90
Managing the Access Gateway Cluster Configuration . . . 90
Protecting Web Resources Through the Access Gateway . . . 92
Configuration Options . . . 92
Managing Reverse Proxies and Authentication . . . 94
Configuring Web Servers of a Proxy Service . . . 100
Configuring Protected Resources . . . 101
Configuring HTML Rewriting . . . 113
Configuring Connection and Session Limits . . . 130
Protecting Multiple Resources . . . 134
Configuring Trusted Providers for Single Sign-On . . . 144
Understanding the Trust Model . . . 144
Configuring General Provider Settings . . . 146
Managing Trusted Providers . . . 150
Modifying a Trusted Provider . . . 154
Communication Security . . . 155
Selecting Attributes for a Trusted Provider . . . 155
Managing Metadata . . . 158
Configuring an Authentication Response for a Service Provider . . . 162
Routing to an External Identity Provider Automatically . . . 162
Configuring Options for Trusted Service Providers . . . 163
Using the Intersite Transfer Service . . . 164
Configuring Single Sign-On to Specific Applications . . . 174
Configuring Protected Resource for a SharePoint Server . . . 174
Configuring a Protected Resource for a SharePoint Server with an ADFS Server . . . 174
Configuring a Protected Resource for Outlook Web Access . . . 177
Configuring a Protected Resource for a Novell Vibe 3.3 Server . . . 180
Configuring Access to the Filr Site through Access Manager . . . 185
Sample Configuration for Protecting an Application Through Access Manager Appliance . . . 185
Installation Overview and Prerequisites . . . 185
Accessing the Sample Web Portal . . . 187
Understanding the Policies Used in the Sample Portal . . . 187

4 Setting Up an Advanced Access Manager Configuration 189

Identity Server Advanced Configuration . . . 189
Managing an Identity Server . . . 189
Editing Server Details . . . 192
Customizing The Identity Server . . . 192
Access Gateway Server Advance Configuration . . . 231
Configuration Overview . . . 231
Saving, Applying, or Canceling Configuration Changes . . . 232
Managing Access Gateways . . . 234
Managing General Details of the Access Gateway . . . 238
Setting Up a Tunnel . . . 240
Setting the Date and Time . . . 241
Configuring Network Settings . . . 242
Enabling the Access Gateway to Display Post-Authentication Message . . . 246

Customizing The Access Gateway . . . 246
Access Gateway Content Settings . . . 252
Configuring Caching Options . . . 253
Controlling Browser Caching . . . 254
Configuring a Pin List . . . 254
Configuring a Purge List . . . 257
Purging Cached Content . . . 258
Apache htcacheclean Tool . . . 258
Access Gateway Advanced Options . . . 259
Configuring the Global Advanced Options . . . 259
Configuring the Advanced Options for a Domain-Based and Path-Based Multi-Homing Proxy Service . . . 269
Modifying Configuration Files . . . 269
Modifying web.xml . . . 269
Modifying server.xml . . . 270

5 Configuring Authentication 271

Configuring Local Authentication . . . 271
Configuring Identity User Stores . . . 272
Creating Authentication Classes . . . 283
Configuring Authentication Methods . . . 287
Configuring Authentication Contracts . . . 288
Specifying Authentication Defaults . . . 296
Social Authentication . . . 298
Two-Factor Authentication Using Time-Based One-Time Password (TOTP) . . . 304
Persistent Authentication . . . 307
RADIUS Authentication . . . 309
Client Integrity Check . . . 310
Mutual SSL (X.509) Authentication . . . 316
ORed Credential Class . . . 321
OpenID Authentication . . . 322
Password Retrieval . . . 323
Configuring Access Manager for NESCM . . . 325
Kerberos Authentication . . . 330
Risk-based Authentication . . . 341
Managing Direct Access to the Identity Server . . . 367
Configuring Federated Authentication . . . 372
Configuring Federation . . . 372
Service Provider Brokering . . . 392
Configuring User Identification Methods for Federation . . . 411
Configuring SAML 2.0 . . . 418
Configuring SAML 1.1 . . . 452
Configuring Liberty . . . 455
Configuring Liberty Web Services . . . 461
Configuring WS Federation . . . 481
Configuring WS-Trust Security Token Service . . . 506
Configuring OAuth and OpenID Connect . . . 527
Configuring Authentication Through Federation for Specific Providers . . . 563
Integrating Amazon Web Services with Access Manager . . . 567
Configuring Single Sign-On for Office 365 Services . . . 570

6 Access Manager Policies 591

Understanding Policies . . . 591
Selecting a Policy Type . . . 592
Tuning the Policy Performance . . . 592
Managing Policies . . . 593
Managing Policy Containers . . . 595

Managing a Rule List . . . 596
Adding Policy Extensions . . . 598
Enabling Policy Logging . . . 602
Role Policies . . . 603
Understanding RBAC in Access Manager Appliance . . . 603
Enabling Role-Based Access Control . . . 607
Creating Roles . . . 608
Example Role Policies . . . 627
Creating Access Manager Appliance Roles in an Existing Role-Based Policy System . . . 630
Mapping Roles between Trusted Providers . . . 640
Enabling and Disabling Role Policies . . . 642
Importing and Exporting Role Policies . . . 642
Authorization Policies . . . 642
Designing an Authorization Policy . . . 643
Creating Access Gateway Authorization Policies . . . 653
Sample Access Gateway Authorization Policies . . . 655
Conditions . . . 662
Importing and Exporting Authorization Policies . . . 691
Identity Injection Policies . . . 691
Designing an Identity Injection Policy . . . 692
Configuring an Identity Injection Policy . . . 693
Configuring an Authentication Header Policy . . . 694
Configuring a Custom Header Policy . . . 698
Configuring a Custom Header with Tags . . . 701
Specifying a Query String for Injection . . . 703
Injecting into the Cookie Header . . . 706
Configuring an Inject Kerberos Ticket Policy . . . 706
Importing and Exporting Identity Injection Policies . . . 709
Sample Identity Injection Policy . . . 709
Form Fill Policies . . . 711
Understanding an HTML Form . . . 711
Creating a Form Fill Policy for the Sample Form . . . 714
Implementing Form Fill Policies . . . 716
Creating and Managing Shared Secrets . . . 732
Importing and Exporting Form Fill Policies . . . 734
Configuring a Form Fill Policy for Forms With Scripts . . . 735
External Attribute Source Policies . . . 739
Enabling External Attributes Policy . . . 739
Creating an External Attribute Source Policy . . . 740
External Attribute Source Policy Examples . . . 740
Risk-Based Policies . . . 745
Configuring Risk-Based Authentication . . . 745
Configuring an Authorization Policy to Protect a Resource . . . 756
Enabling Auditing for Risk-Based Authentication Events . . . 756
Enabling Logging for Risk-Based Authentication . . . 756
Risk-Based Authentication: Sample Configuration . . . 757

7 Enabling Mobile and Web Access 761

User Requirements for MobileAccess . . . 761
Configuring Appmarks . . . 762
Creating Multiple Appmarks for an Application . . . 762
Understanding Appmark Options . . . 763
Configuring MobileAccess . . . 764
Helping Users Register Their Mobile Devices . . . 765
Registering iOS Devices . . . 765
Registering Android Devices . . . 766
Installing MobileAccess on a Mobile Device . . . 766

Understanding the MobileAccess PIN . . . 767
Managing Mobile Devices . . . 767
Deregistering Mobile Devices . . . 767
Deregistering a Mobile Device as a User . . . 768
Deleting and Reinstalling the MobileAccess App on a Device . . . 768
Installing the Access Manager Certificate Authority Certificate on a Mobile Device . . . 768
Changing the Branding of the User Portal Page . . . 769

8 High Availability and Fault Tolerance 771

Installing Secondary Versions of Access Manager Appliance . . . 771
Prerequisites . . . 771
Understanding How Consoles Interact with Each Other and with Access Manager Devices . . . 773
Configuration Tips for the L4 Switch . . . 774
Sticky Bit . . . 774
Network Configuration Requirements . . . 774
Health Checks . . . 775
Real Server Settings Example . . . 778
Virtual Server Settings Example . . . 779
Setting up L4 Switch for IPv6 Support . . . 779
Web SSO Over IPv6 . . . 780
Federated SSO over IPv6 . . . 781
Limitations . . . 782
Using a Software Load Balancer . . . 783

Part II Security 785

9 Securing Access Manager 787

Securing the Administration Console . . . 787
Protecting the Configuration Store . . . 788
Security Considerations for Certificates . . . 788
Configuring Secure Communication on the Identity Server . . . 789
Viewing the Services That Use the Signing . . . 789
Viewing Services That Use the Encryption . . . 790
Enabling Secure Cookies . . . 790
Securing the Embedded Service Provider Session Cookie on the Access Gateway . . . 791
Securing the Proxy Session Cookie . . . 792
Preventing Cross-site Scripting Attacks . . . 792
Option 1: HTML Escaping . . . 793
Option 2: Filtering . . . 793

10 Understanding Access Manager Certificates 797

Process Flow . . . 798

11 Creating Certificates 799

Creating a Locally Signed Certificate . . . 799
Editing the Subject Name . . . 801
Assigning Alternate Subject Names . . . 803
Generating a Certificate Signing Request . . . 804
Importing a Signed Certificate . . . 805

12 Managing Certificates and Keystores 807

Viewing Certificate Details . . . 807
Renewing a Certificate . . . 809
Exporting a Private/Public Key Pair . . . 811
Exporting a Public Certificate . . . 811
Importing a Private/Public Key Pair . . . 812

13 Assigning Certificates to Access Manager Appliance 813

14 Managing Trusted Roots and Trust Stores 815

Managing Trusted Roots . . . 815
Importing Public Key Certificates (Trusted Roots) . . . 815
Auto-Importing Certificates from Servers . . . 816
Exporting the Public Certificate of a Trusted Root . . . 816
Viewing Trusted Root Details . . . 816
Viewing External Trusted Roots . . . 817

15 Enabling SSL Communication 819

Enabling SSL Communication . . . 819
Using Access Manager Certificates . . . 819
Using Externally Signed Certificates . . . 822
SSL Renegotiation . . . 825
Using SSL on the Access Manager Appliance Communication Channels . . . 826
Prerequisites for SSL . . . 827
Prerequisites for SSL Communication between the Identity Server and Access Manager Appliance . . . 827
Prerequisites for SSL Communication between the Access Gateway and Web Servers. . . . 828
Configuring SSL Communication with Browsers and the Identity Server . . . 828
Configuring SSL between the Proxy Service and the Web Servers . . . 828
Configuring the SSL Communication . . . 829

Part III Maintaining Access Manager 831

16 Auditing 833

Enabling Auditing . . . 834
Specifying the Logging Server and Console Events . . . 835
Configuring Syslog for Auditing . . . 836
Enabling Identity Server Audit Events . . . 837
Enabling Access Gateway Audit Events . . . 840

17 Reporting 843

Overview . . . 843
Prerequisites . . . 844
Deploying Access Manager Reporting Solution Pack . . . 845
Enabling Reporting . . . 845
Generating Reports . . . 846

18 Logging 847

Understanding the Types of Logging . . . 847

Component Logging for Troubleshooting Configuration or Network Problems . . . 847
HTTP Transaction Logging for Proxy Services . . . 848
Understanding the Log Format . . . 849
Understanding the Correlation Tags in the Log Files . . . 850
Sample Scenario . . . 851
Identity Server Logging . . . 852
Configuring Logging for Identity Server . . . 852
Configuring Session-Based Logging . . . 854
Access Gateway Logging . . . 860
Managing Access Gateway Logs . . . 860
Configuring Logging for a Proxy Service . . . 861
Downloading Log Files . . . 869
Administration Console Logs . . . 870
Identity Server Logs . . . 870
Access Gateway Appliance and Access Gateway Service Logs . . . 870
Turning on Logging for Policy Evaluation . . . 871

19 Monitoring Component Statistics 873

Identity Server Statistics . . . 873
Monitoring Identity Server Statistics . . . 873
Monitoring Identity Server Cluster Statistics . . . 881
Access Gateway Statistics . . . 882
Monitoring Access Gateway Statistics . . . 882
Monitoring Access Gateway Cluster Statistics . . . 891
Component Statistics Through REST APIs . . . 893
Monitoring API for the Identity Server Statistics . . . 893
Monitoring API for the Access Gateway Statistics . . . 898

20 Monitoring Server Health 901

Health States . . . 901
Monitoring Health by Using the Hardware IP Address . . . 902
Monitoring Health of Identity Servers . . . 902
Monitoring the Health of an Identity Server . . . 902
Monitoring the Health of a Cluster . . . 904
Monitoring the Health of Access Gateways . . . 904
Monitoring the Health of an Access Gateway . . . 904
Monitoring the Health of an Access Gateway Cluster . . . 906

21 Monitoring Component Command Status 907

Viewing the Command Status of the Identity Server . . . 907
Viewing the Status of Current Commands . . . 907
Viewing Detailed Command Information . . . 908
Viewing the Command Status of the Access Gateway . . . 908
Viewing the Status of Current Commands . . . 908
Viewing Detailed Command Information . . . 909
Reviewing the Command Status for Certificates . . . 910

22 Monitoring Alerts 911

Monitoring Identity Server Alerts . . . 911
Monitoring Access Gateway Alerts . . . 911
Viewing Access Gateway Alerts . . . 911
Viewing Access Gateway Cluster Alerts . . . 912
Managing Access Gateway Alert Profiles . . . 912

Configuring an Alert Profile . . . 912
SNMP Profile . . . 914
Configuring a Log Profile . . . 914
Configuring an E-Mail Profile . . . 914
Configuring a Syslog Profile . . . 915

23 Monitoring Access Manager By Using Simple Network Management Protocol 917

SNMP Architecture in Access Manager . . . 917
Features of Monitoring in Access Manager . . . 918
Using the Default MIB File with External SNMP Systems . . . 919
Querying For SNMP Attributes . . . 920
Querying Using the Namespace . . . 921
Querying Using the OID . . . 921
Installing and Enabling Monitoring for Access Manager Components . . . 921
Installing and Enabling Monitoring for Access Manager on Linux . . . 922
Installing and Enabling Monitoring for Access Manager on Windows . . . 922

24 Back Up and Restore 925

How The Backup and Restore Process Works . . . 925
Default Parameters . . . 925
The Process . . . 925
Backing Up the Access Manager Appliance Configuration . . . 926
Restoring the Access Manager Appliance Configuration . . . 927
Restoring the Configuration on the Same Appliance for Which Backup Was Taken . . . 928
Restoring the Configuration on a Freshly Installed Appliance with Same IP Address and DNS Settings . . . 928

25 Code Promotion 931

How Code Promotion Helps . . . 931
Sequence of Promoting the Configuration Data . . . 932
Prerequisites . . . 932
Limitations . . . 933
Configuring Custom File Paths . . . 933
Exporting the Configuration Data . . . 934
Importing the Configuration Data . . . 935
Uploading Configuration File to Import . . . 935
Selecting the Component to Import the Configuration Data . . . 936
Importing the Identity Server Configuration Data . . . 936
Importing the Access Gateway Configuration Data . . . 937
Post-Import Configuration Tasks . . . 940
Troubleshooting Code Promotion . . . 942

26 Troubleshooting 943

Troubleshooting Installation . . . 943
Checking the Installation Logs . . . 943
Some of the New Hardware Drivers or Network Cards Are Not Detected during Installation . . . 944
Installation Through Terminal Mode is not Supported . . . 944
Novell Device Manager Installation Fails During the Appliance Installation . . . 945
Access Manager Appliance Installation Fails Due to an XML Parser Error . . . 945
DN Is Added as Provider ID While Installing NMAS SAML Method . . . 945
Troubleshooting Upgrade . . . 945
The Access Gateway Throws a 403 Forbidden Page Error for a Resource Protected by a Form Fill Policy . . . 945

DN Is Added as Provider ID While Installing NMAS SAML Method . . . 946
Issue in SSL Communication between the Access Gateway and Web Applications . . . 946
Customized Login Pages Are Missing After Upgrading Access Manager . . . 946
Troubleshooting the Administration Console . . . 946
Global Troubleshooting Options . . . 947
Diagnostic Configuration Export Utility . . . 950
Logging . . . 951
Restoring a Failed Secondary Console . . . 951
Converting a Secondary Access Manager Appliance into a Primary Appliance . . . 952
Repairing the Configuration Datastore . . . 956
Session Conflicts . . . 956
Unable to Log In to the Administration Console . . . 957
Exception Processing IdentityService_ServerPage.JSP . . . 957
Backup and Restore Failure Because of Special Characters in Passwords . . . 957
Unable to Install NMAS SAML Method . . . 958
Incorrect Audit Configuration . . . 958
Unable to Update the Access gateway Listening IP Address in the Administration Console Reverse Proxy . . . 959
During Access Manager Appliance Installation Any Error Message Should Not Display Successful Status . . . 960
Incorrect Health Is Reported on the Access Gateway . . . 960
Administration Console Does Not Refresh the Command Status Automatically . . . 960
SSL Communication with Weak Ciphers Fails . . . 960
Error: Tomcat did not stop in time. PID file was not removed . . . 961
An IP Address for the Other Known Device Manager List is Missing in the Troubleshooting Page . . . 961
Troubleshooting the Access Gateway . . . 961
Useful Troubleshooting Files . . . 962
Verifying That All Services Are Running . . . 964
Troubleshooting SSL Connection Issues . . . 966
Enabling Debug Mode and Core Dumps . . . 966
Useful Troubleshooting Tools for the Access Gateway Service . . . 968
Solving Apache Restart Issues . . . 968
Understanding the Authentication Process of the Access Gateway Service . . . 970
Enabling Caching of Audit Events for Apache Gateway Service . . . 977
Issue While Accelerating the Ajax Applications . . . 978
Accessing Lotus-iNotes through the Access Gateway Asks for Authentication . . . 978
Configuration Issues . . . 978
Cannot Inject a Photo into HTTP Headers . . . 978
Access Gateway Caching Issues . . . 979
Issues while Changing Management IP Address on an Access Gateway Appliance . . . 979
Issue while Adding the Access Gateway in a Cluster . . . 980
Troubleshooting Identity Server and Authentication . . . 981
Useful Networking Tools for Linux Identity Server . . . 982
Troubleshooting 100101043 and 100101044 Liberty Metadata Load Errors . . . 982
Authentication Issues . . . 989
After Setting Up the User Store to Use SecretStore, Users Report 500 Errors . . . 991
When Multiple Browser Logout Option Is Enabled, User Is Not Getting Logged Out from Different Sessions . . . 991
After Consuming a SAML Response, the Browser Is redirected to an Incorrect URL . . . 992
Configuring SAML 1.1 Identity Provider Without Specifying Port in the Login URL Field . . . 992
Attributes Are Not Available Through Form Fill When OIOSAML Is Enabled . . . 992
Issue in Importing Metadata While Configuring Identity Provider or Service Provider Using Metadata URL . . . 992
Metadata Mentions Triple Des As Encryption Method . . . 993
Issue in Accessing Protected Resources with External Identity Provider When Both Providers Use Same Cookie Domain . . . 993
SAML Intersite Transfer URL Setup Does Not Work for Non-brokered Setups after Enabling SP Brokering . . . 993
Orphaned Identity Objects . . . 993

Users cannot Log In to Identity Provider When They Access Protected Resources With Any Contract Assigned . . . 994
Attribute Query from OIOSAML.SP Java Service Provider Fails with Null Pointer . . . 994
Disabling the Certificate Revocation List Checking . . . 994
Step up Authentication for the Identity Server Initiated SSO to External Provider Does not Work
Unless It has a Matching Local Contract . . . 995
Metadata Cannot be Retrieved from the URL . . . 995
Requesting the Authentication to a Service Provider Fails . . . 995
SAML 2.0 POST Compression Failure Does not Throw a Specific Error Code . . . 995
SAML 1.1 Service Provider Re-requests for Authentication . . . 995
The Identity Server Statistics Logs Do Not Get Written In Less Than One Minute . . . 996
No Error Message Is Written in the Log File When an Expired Certificate Is Used for the X509 Authentication . . . 996
Terminating an Existing Authenticated User from the Identity Server . . . 996
X.509 Authentication Lists the Entire List of Certificates Imported to the Browser . . . 997
Clustered Nodes Looping Due to JGroup Issues . . . 997
Authentication With Aliases Fails . . . 998
nidp/app Does Not Redirect to nidp/portal after Authentication . . . 998
Login to Office 365 Fails when WS-Trust MEX Metadata Is Larger than 65 KB . . . 998
Unsafe Server Certificate Change in SSL/TLS Renegotiations Is Not Allowed . . . 999
Viewing Request and Response Headers of All Protocols in a Log File . . . 999
Troubleshooting Certificate Issues . . . 1000
Resolving Certificate Import Issues . . . 1000
Mutual SSL with X.509 Produces Untrusted Chain Messages . . . 1003
Certificate Command Failure . . . 1003
A Device Reports Certificate Errors . . . 1003
Renewing the expired eDirectory certificates . . . 1003
Troubleshooting Access Manager Policies . . . 1004
Turning on Logging for Policy Evaluation . . . 1004
Common Configuration Problems That Prevent a Policy from Being Applied as Expected . . . 1005
The Policy Is Using Old User Data . . . 1007
Form Fill and Identity Injection Silently Fail . . . 1009
Checking for Corrupted Policies . . . 1009
Policy Page Timeout . . . 1009
Policy Creation and Storage . . . 1009
Policy Distribution . . . 1010
Policy Evaluation: Access Gateway Devices . . . 1011
Troubleshooting MobileAccess . . . 1015
Using the Same Mobile Device for Different Users Causes the Expired Session Error . . . 1015
Simple Authentication with a Pop-up Browser Window Does Not Work for MobileAccess . . . 1015
Users Fail to Authenticate to MobileAccess when Appmarks Are Launched in the Chrome Browser . . . 1015
Troubleshooting Code Promotion . . . 1016
Troubleshooting Identity Server Code Promotion . . . 1016
Troubleshooting Access Gateway Code Promotion . . . 1016
Troubleshooting Device Customization Code Promotion . . . 1019
Troubleshooting OAuth and OpenID Connect . . . 1020
Users Cannot Register a Client Application . . . 1020
Token Exchanges Show Redirect URI Invalid Error . . . 1020
Users Cannot Register or Modify a Client Application with Specific Options . . . 1020
A Specific Claim Does Not Come to the UserInfo Endpoint during Claims Request . . . 1021
Access Gateway OAuth Fails . . . 1021
After Allowing Consent, 500 Internal Server Error Occurs . . . 1021
No Error Message When a Token Request Contains Repetitive Parameters . . . 1021
Oauth Token Encryption/Signing Key Is Compromised or Corrupted . . . 1021
Tracing OAuth Requests . . . 1022
OAuth Client Registration Fails If a Role Policy Contains a Condition Other than LDAP Attribute, LDAP Group, or LDAP OU . . . 1022

The Identity Injection Policy Does Not Inject Passwords . . . 1023
OAuth Endpoint Generates Access and Refresh Tokens Using OAuth Resource Owner Credential Flow . . . 1023
OAuth Apps Fail After Upgrading Access Manager . . . 1023
Troubleshooting User Attribute Retrieval and Transformation . . . 1023
In Identity Provider No Value is Fetched from Attribute Source . . . 1023
Error Message While Testing a Database Connection . . . 1024
Regex Replace Error Message . . . 1024
Using Log Files for Troubleshooting . . . 1024
Sample Authentication Traces . . . 1025
Understanding Policy Evaluation Traces . . . 1028
Access Manager Audit Events and Data . . . 1045
NIDS: Sent a Federate Request (002e0001) . . . 1049
NIDS: Received a Federate Request (002e0002) . . . 1049
NIDS: Sent a Defederate Request (002e0003) . . . 1050
NIDS: Received a Defederate Request (002e0004) . . . 1050
NIDS: Sent a Register Name Request (002e0005) . . . 1051
NIDS: Received a Register Name Request (002e0006) . . . 1051
NIDS: Logged Out an Authentication that Was Provided to a Remote Consumer (002e0007) . . . 1052
NIDS: Logged out a Local Authentication (002e0008) . . . 1052
NIDS: Provided an Authentication to a Remote Consumer (002e0009) . . . 1053
NIDS: User Session Was Authenticated (002e000a) . . . 1053
NIDS: Failed to Provide an Authentication to a Remote Consumer (002e000b) . . . 1054
NIDS: User Session Authentication Failed (002e000c) . . . 1054
NIDS: Received an Attribute Query Request (002e000d) . . . 1055
NIDS: User Account Provisioned (002e000e) . . . 1055
NIDS: Failed to Provision a User Account (002e000f) . . . 1056
NIDS: Web Service Query (002e0010) . . . 1056
NIDS: Web Service Modify (002e0011) . . . 1057
NIDS: Connection to User Store Replica Lost (002e0012) . . . 1057
NIDS: Connection to User Store Replica Reestablished (002e0013) . . . 1058
NIDS: Server Started (002e0014) . . . 1058
NIDS: Server Stopped (002e0015) . . . 1059
NIDS: Server Refreshed (002e0016) . . . 1059
NIDS: Intruder Lockout (002e0017) . . . 1060
NIDS: Severe Component Log Entry (002e0018) . . . 1060
NIDS: Warning Component Log Entry (002e0019) . . . 1061
NIDS: Failed to Broker an Authentication from Identity Provider to Service Provider as Identity Provider and Service Provider Are not in Same Group (002E001A) . . . 1061
NIDS: Failed to Broker an Authentication from Identity Provider to Service Provider Because a Policy Evaluated to Deny (002E001B) . . . 1062
NIDS: Brokered an Authentication from Identity Provider to Service Provider (002E001C) . . . 1062
NIDS: OAuth2 Authorization code issued (002e0028) . . . 1063
NIDS: OAuth2 token issued (002e0029) . . . 1063
NIDS: OAuth2 Authorization code issue failed (002e0030) . . . 1064
NIDS: OpenID token issued (002e0031) . . . 1064
NIDS: OAuth2 refresh token issued (002e0032) . . . 1065
NIDS: OAuth2 token issue failed (002e0033) . . . 1065
NIDS: OpenID token issue failed (002e0034) . . . 1066
NIDS: OAuth2 refresh token issue failed (002e0035) . . . 1066
NIDS: OAuth2 client has been registered successfully (002e0036) . . . 1066
NIDS: OAuth2 client has been modified successfully (002e0037) . . . 1067
NIDS: OAuth2 client has been deleted successfully (002e0038) . . . 1067
NIDS: OAuth2 user has provided consent (002e0039) . . . 1068
NIDS: OAuth2 user has revoked consent (002e0040) . . . 1068
NIDS: OAuth2 token validation success (002e0041) . . . 1069
NIDS: OAuth2 token validation failed (002e0042) . . . 1069
NIDS: OAuth2 client registration failed (002e0043) . . . 1070
NIDS: Roles PEP Configured (002e0300) . . . 1070
Access Gateway: PEP Configured (002e0301) . . . 1070

Roles Assignment Policy Evaluation (002e0320) . . . 1071
Access Gateway: Authorization Policy Evaluation (002e0321) . . . 1071
Access Gateway: Form Fill Policy Evaluation (002e0322) . . . 1072
Access Gateway: Identity Injection Policy Evaluation (002e0323) . . . 1072
Access Gateway: Access Denied (0x002e0505) . . . 1073
Access Gateway: URL Not Found (0x002e0508) . . . 1073
Access Gateway: System Started (0x002e0509) . . . 1074
Access Gateway: System Shutdown (0x002e050a) . . . 1075
Access Gateway: Identity Injection Parameters (0x002e050c) . . . 1075
Access Gateway: Identity Injection Failed (0x002e050d) . . . 1076
Access Gateway: Form Fill Authentication (0x002e050e) . . . 1076
Access Gateway: Form Fill Authentication Failed (0x002e050f) . . . 1077
Access Gateway: URL Accessed (0x002e0512) . . . 1078
Access Gateway: IP Access Attempted (0x002e0513) . . . 1078
Access Gateway: Webserver Down (0x002e0515) . . . 1079
Access Gateway: All WebServers for a Service is Down (0x002e0516) . . . 1079
Management Communication Channel: Health Change (0x002e0601) . . . 1080
Management Communication Channel: Device Imported (0x002e0602) . . . 1081
Management Communication Channel: Device Deleted (0x002e0603) . . . 1081
Management Communication Channel: Device Configuration Changed (0x002e0604) . . . 1082
Management Communication Channel: Device Alert (0x002e0605) . . . 1082
Risk-Based Authentication: 002e0025 . . . 1083
Risk-Based Authentication: 002e0026 . . . 1083
Risk-Based Authentication: 002e0027 . . . 1084
Event Codes . . . 1085
Administration Console (009) . . . 1085
Identity Server (001) . . . 1122
Linux Access Gateway Appliance(045) . . . 1163
Access Gateway Service (046) . . . 1164
Policy Engine (008) . . . 1168
SOAP Policy Enforcement Point (011) . . . 1173
Backup and Restore (010) . . . 1178
NetIQ Modular Authentication Class (012) . . . 1183

Part IV Appendix 1185

A Data Model Extension XML 1187

Elements . . . 1187
Writing Data Model Extension XML . . . 1190

B SOAP versus REST API 1193

C OAuth versus Other Protocols 1195

D Access Manager Reports Samples 1197

Application Access Summary Report . . . 1198
User Application Access Summary Report . . . 1199
Application Specific User Access Report . . . 1200
Federation Summary Report . . . 1201
User Login Contract Summary Report . . . 1202
User Login Failure Report . . . 1203
Application Specific Risk based Authentication Report . . . 1204

About NetIQ Corporation

We are a global, enterprise software company, with a focus on the three persistent challenges in your environment: Change, complexity and risk—and how we can help you control them.

Our Viewpoint

Adapting to change and managing complexity and risk are nothing new

In fact, of all the challenges you face, these are perhaps the most prominent variables that deny you the control you need to securely measure, monitor, and manage your physical, virtual, and cloud computing environments.

Enabling critical business services, better and faster

We believe that providing as much control as possible to IT organizations is the only way to enable timelier and cost effective delivery of services. Persistent pressures like change and complexity will only continue to increase as organizations continue to change and the technologies needed to manage them become inherently more complex.

Our Philosophy

Selling intelligent solutions, not just software

In order to provide reliable control, we first make sure we understand the real-world scenarios in which IT organizations like yours operate — day in and day out. That's the only way we can develop practical, intelligent IT solutions that successfully yield proven, measurable results. And that's so much more rewarding than simply selling software.

Driving your success is our passion

We place your success at the heart of how we do business. From product inception to deployment, we understand that you need IT solutions that work well and integrate seamlessly with your existing investments; you need ongoing support and training post-deployment; and you need someone that is truly easy to work with — for a change. Ultimately, when you succeed, we all succeed.

Our Solutions

- Identity & Access Governance
- Access Management
- Security Management
- Systems & Application Management
- Workload Management
- Service Management

Contacting Sales Support

For questions about products, pricing, and capabilities, contact your local partner. If you cannot contact your partner, contact our Sales Support team.

Worldwide:	www.netiq.com/about_netiq/officelocations.asp
United States and Canada:	1-888-323-6768
Email:	info@netiq.com
Web Site:	www.netiq.com

Contacting Technical Support

For specific product issues, contact our Technical Support team.

Worldwide:	www.netiq.com/support/contactinfo.asp
North and South America:	1-713-418-5555
Europe, Middle East, and Africa:	+353 (0) 91-782 677
Email:	support@netiq.com
Web Site:	www.netiq.com/support

Contacting Documentation Support

Our goal is to provide documentation that meets your needs. The documentation for this product is available on the NetIQ web site in HTML and PDF formats on a page that does not require you to log in. If you have suggestions for documentation improvements, click **comment on this topic** at the bottom of any page in the HTML version of the documentation posted at www.netiq.com/documentation. You can also email Documentation-Feedback@netiq.com. We value your input and look forward to hearing from you.

Contacting the Online User Community

NetIQ Communities, the NetIQ online community, is a collaborative network connecting you to your peers and NetIQ experts. By providing more immediate information, useful links to helpful resources, and access to NetIQ experts, NetIQ Communities helps ensure you are mastering the knowledge you need to realize the full potential of IT investments upon which you rely. For more information, visit http://community.netiq.com.

About this Book and the Library

The *Administration Guide* provides an introduction to NetIQ Access Manager Appliance and details about how to configure Access Manager features.

Intended Audience

This book is intended for Access Manager administrators. It is assumed that you have knowledge of evolving Internet protocols, such as:

- Extensible Markup Language (XML)
- Simple Object Access Protocol (SOAP)
- Security Assertion Markup Language (SAML)
- Public Key Infrastructure (PKI) digital signature concepts and Internet security
- Secure Socket Layer/Transport Layer Security (SSL/TLS)
- Hypertext Transfer Protocol (HTTP and HTTPS)
- Uniform Resource Identifiers (URIs)
- Domain Name System (DNS)
- Web Services Description Language (WSDL)

Other Information in the Library

The library provides the following information resources:

Installation and Upgrade Guide

Provides an introduction to NetIQ Access Manager Appliance and describes the installation and upgrade procedures.

Help

Provides context-sensitive information and step-by-step guidance for common tasks, as well as definitions for each field on each window.

Developer Documentation

Provides a collection of developer tools and examples to design a flexible and expandable access management system to enable your applications to interact with the Identity Management capabilities of Access Manager, including federation, provisioning, and the secure delivery of identity information to client-based applications.

NOTE: Contact namsdk@netiq.com for any query related to Access Manager SDK.

---

1 Overview

NetIQ Access Manager Appliance is a comprehensive access management solution that provides secure access to web applications. Access Manager also provides seamless single sign-on across technical and organizational boundaries. It uses industry standards including Secure Assertions Markup Language (SAML) and Liberty Alliance protocols. It has a single console for management and configuration. To provide secure access from any location or any device, it supports multi-factor authentication, role-based access control, and data encryption.

This section discusses the following topics:

- "How Access Manager Appliance Solves Business Challenges" on page 21
- "How Access Manager Appliance Works" on page 28
- "Access Manager Appliance Components and Their Features" on page 30
- "Language Support" on page 33

How Access Manager Appliance Solves Business Challenges

As networks expand to connect people and businesses throughout the world, secure access to business resources becomes increasingly important and more complex. Today, employees work from corporate, home, and mobile offices using mobile devices. Besides employees, customers and partners also require access to resources on your network, and your employees require access to resources on partners' networks or at service providers.

Access Manager Appliance lets you provide employees, customers, and partners with secure access to your network resources using any type of device. Access Manager Appliance helps you if your business faces any of the following access-related challenges:

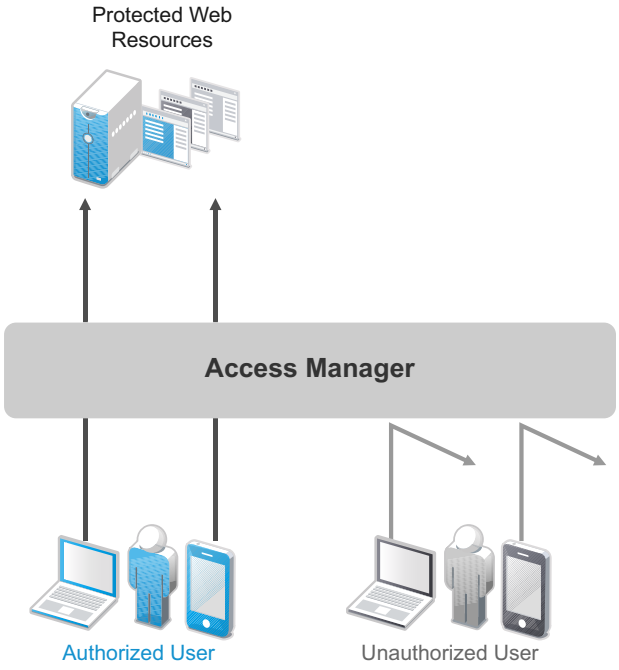
- Protecting resources so that only authorized users can access them whether those users are employees, customers, or partners.
- Ensuring that the users who are authorized to use a resource can access that resource regardless of users' location or type of device they use.
- Requiring users to manage multiple passwords for authentication to Web applications.
- Ensuring that users have access only to the resources required for their jobs.
- Revoking network access from users faster.
- Protecting users' privacy and confidential information as they access company resources or partners' resources.
- Proving compliance with your business policies, privacy laws such as Sarbanes-Oxley, HIPAA, or European Union, and other regulatory requirements.
- Providing facility to access protected resources by using token-based protocols.
- Providing facility to use your existing credentials to access services from different service providers such as Office 365 and Salesforce.

The following sections expand on these challenges and introduce the solutions provided by Access Manager Appliance.

- “Protecting Resources While Providing Access” on page 22
- “Managing Passwords with Single Sign-On” on page 23
- “Enforcing Business Policies” on page 24
- “Sharing Identity Information” on page 25
- “Protecting Identity Information” on page 27
- “Complying with Regulations” on page 28

Protecting Resources While Providing Access

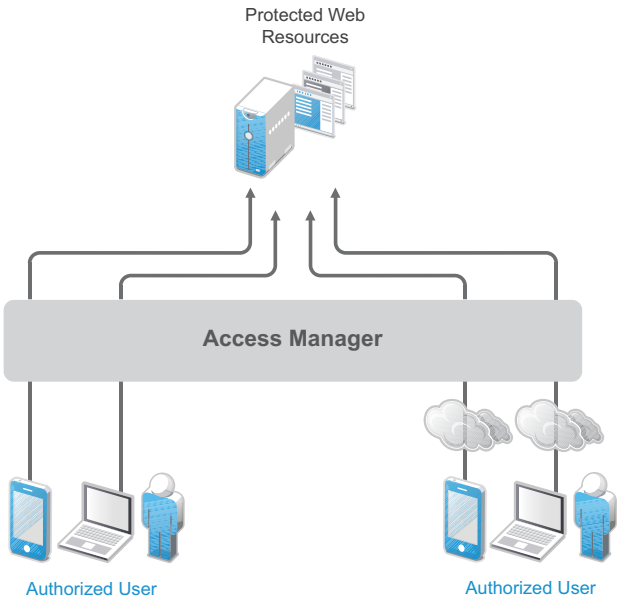
The primary purpose of Access Manager Appliance is to protect resources by allowing access only to users you have authorized. As shown in the following illustration, those users who are authorized to use the protected resources, are allowed access, while unauthorized users are denied access.



Access Manager Appliance secures your web resources from Internet hackers. Addresses of the servers that host the protected resources are hidden from both external and internal users. The only way to access the resources is by logging in to Access Manager Appliance with authorized credentials, no matter what device you use.

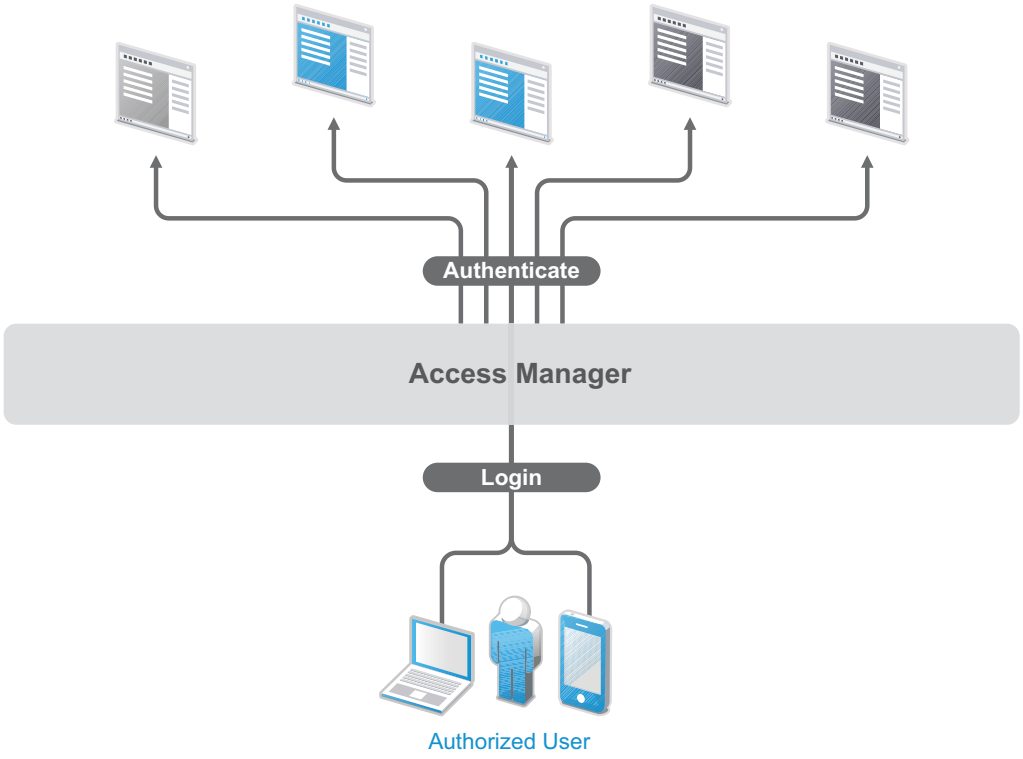
Access Manager Appliance protects only the resources you have set up as protected resources. It is not a firewall and should always be used in conjunction with a firewall product.

Access to resources is independent of a user’s location, as shown in the following illustration. Access Manager Appliance provides the same secure access and same experience whether the user is accessing resources from your local office, from home, or from any other place with any type of device.



Managing Passwords with Single Sign-On

Authentication through Access Manager Appliance not only establishes authorization to applications (see Protecting Resources While Providing Access), but it can also provide authorization to those same applications. With Access Manager Appliance serving as the front-end authentication, you can deploy standards-based Web single sign-on. With single sign-on, your employees, partners, and customers only need to remember one password or login routine to access all corporate and Web-based applications they are authorized to use. That means fewer help desk calls and the reduced likelihood of users resorting to vulnerable written reminders.

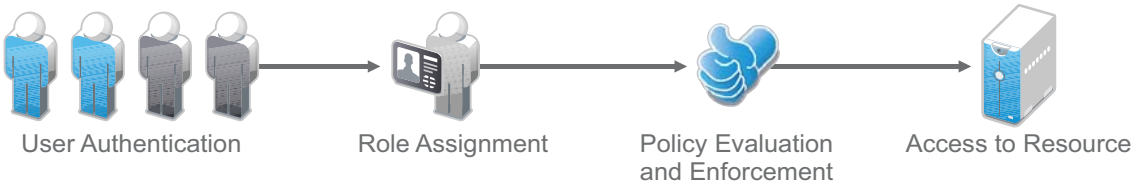


By simplifying the use and management of passwords, Access Manager Appliance helps you enhance users' experience, increase security, streamline business processes, and reduce system administration and support costs.

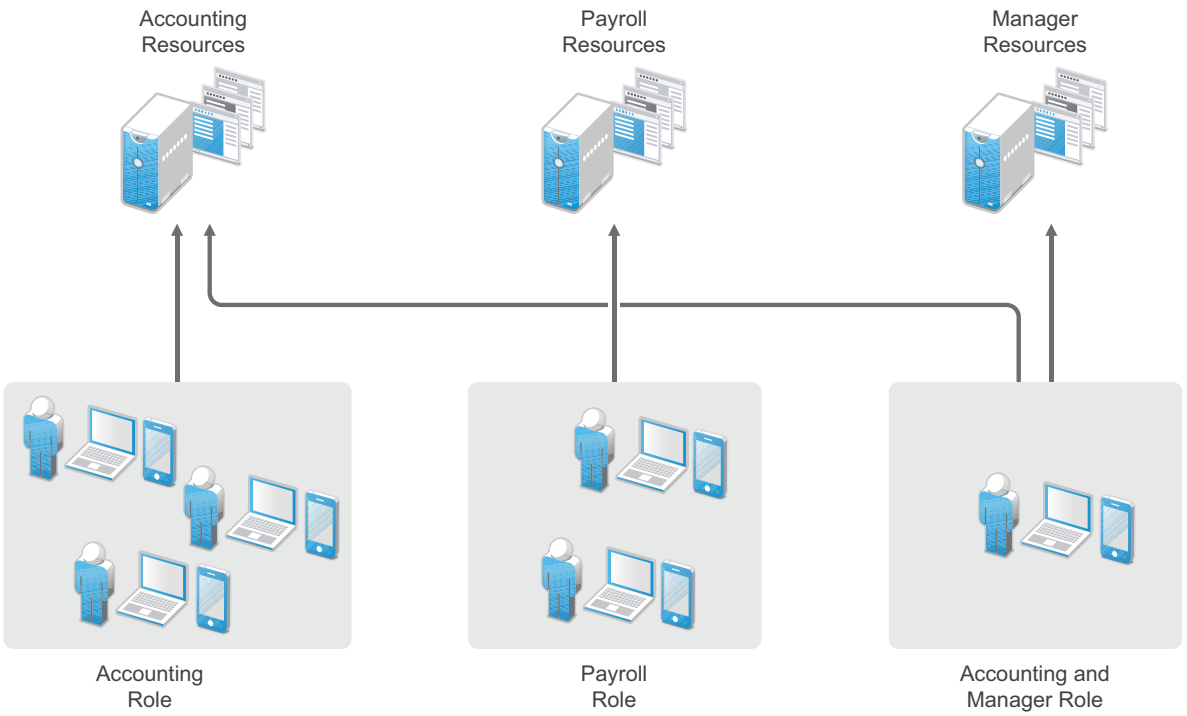
Enforcing Business Policies

Determining the access policies for an organization is often complicated and difficult, but the difficulty pales in comparison to enforcing the policies. Your IT personnel can spend hours attempting to give users the correct access to resources, and hours more retracing their steps to see why authorized users cannot access resources. You might never know about the situations where users access resources they should not be accessing.

Access Manager Appliance automates the granting and revoking of access through the use of roles and policies. As shown in the following illustration, users are assigned to roles that have access policies associated with them. Each time a user authenticates through Access Manager Appliance, the user's access is determined by the policies associated with the user's roles.



In the following example, users assigned to the Accounting role receive access to the Accounting resources, Payroll users receive access to the Payroll resources, and Accounting managers receive access to both the Accounting and Manager resources.



Because access is based on roles, you can grant access in minutes and be certain that the access is consistent with your business policies. You can revoke access in minutes by removing role assignments from users.

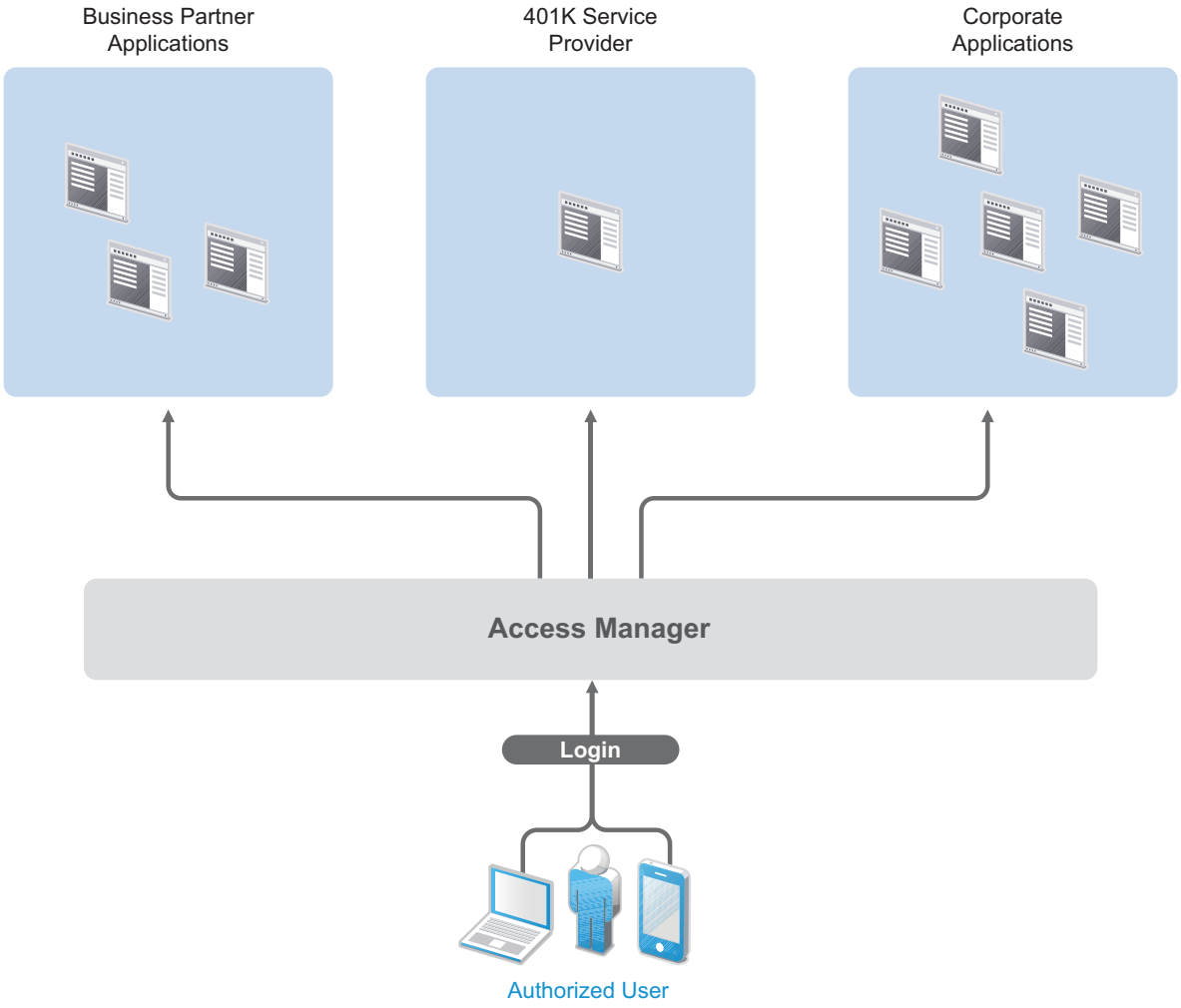
For security-minded organizations, it comes down to this simple fact: you set the policies by which users gain access, and Access Manager Appliance enforces them consistently and quickly. There are no surprises and no delays.

You can also securely grant access to user's private resources such as web applications, mobile phones, handheld devices, and desktop by using access tokens.

Sharing Identity Information

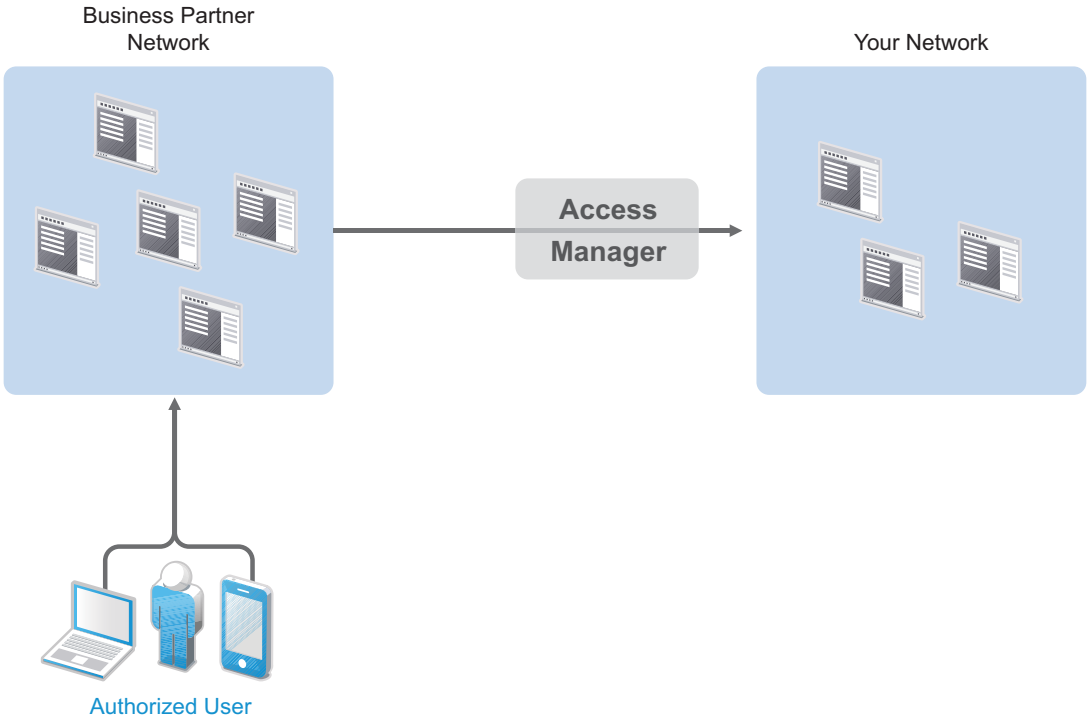
In today's business environment, few organizations stand alone. More than likely, you have trusted business partners with whom you need to share resources in a secure manner. Or, you have business services, such as a 401k management system, to which you need to provide employee access. Or, maybe your organization is the one providing services to another business. Access Manager Appliance provides federated identity management to enable users to seamlessly and securely authenticate across autonomous identity domains.

For example, assume that you have employees who need access to your corporate applications, several business partners' applications, and their 401k service, as shown in the following figure:



Each identity domain (your organization, your partner's organization, and 401k service) requires an account and authentication to that account to access the resources. However, because you have used Access Manager Appliance to establish a trust relationship with the business partner and the 401k service, your employees can log in through Access Manager Appliance to gain access to the authorized resources in all three identity domains.

Access Manager Appliance enables your employees to access resources from business partners and service providers. It also lets business partners access authorized resources on your network. The following figure illustrates this type of access.



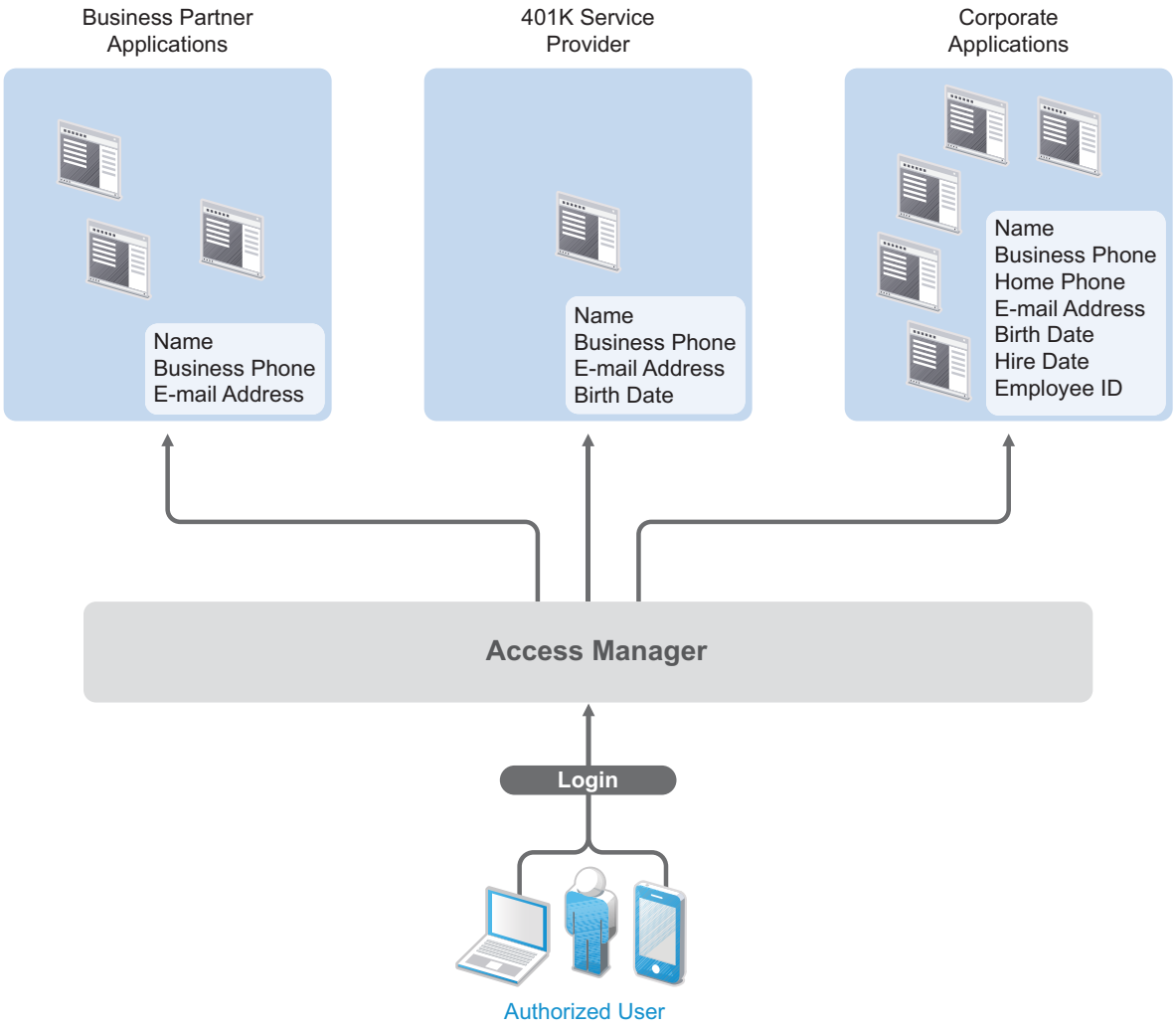
In addition to simply linking user accounts in different identity domains, Access Manager Appliance also supports federated provisioning, which means that new user accounts can be automatically created in your trusted partner's (or provider's) system. For example, a new employee in your organization can initiate the creation of an account in your business partner's system through Access Manager Appliance rather than relying on the business partner to provide the account. Customers or trusted business partners can automatically create accounts in your system.

Access Manager Appliance leverages identity federation standards including Liberty Alliance, WS-Federation, WS-Trust, and SAML. It also provides a facility to identify risk associated with login attempts, mitigate the risk and take action based on risk severity.This foundation minimizes—or even eliminates—interoperability issues among external partners or internal workgroups. In fact, Access Manager Appliance features an identical configuration process for all federation partners whether they are different departments within your organization or external business partners.

Protecting Identity Information

Whenever you exchange identity information with other businesses or service providers, you must be concerned with protecting the privacy of your employees, customers, and partners. It is an integral part of trusted business partnerships and regulatory compliance: the ability to establish policies on the exchange of identity information.

Access Manager Appliance enables you to determine which business and personal information from your corporate directory to share with others. As shown in the following illustration, you can choose to share only the information required to establish the account at the service provider or trusted partner:



Access Manager Appliance offers this built-in privacy protection for your employees, partners, and customers alike, wherever they are working. With Access Manager Appliance in place, your organization can guarantee user confidentiality. For federated provisioning, Access Manager Appliance adheres to those same policies and protections.

Complying with Regulations

Regulations can be a hassle, but an agile, automated IT infrastructure substantially cuts costs and reduces the pain of compliance. By implementing access based on user identities, you can protect users' privacy and confidential information. At the same time, you can reduce the amount of paperwork needed to prove that proper access control measures are in place. Compliance assurance and documentation is an inherent benefit of Access Manager Appliance.

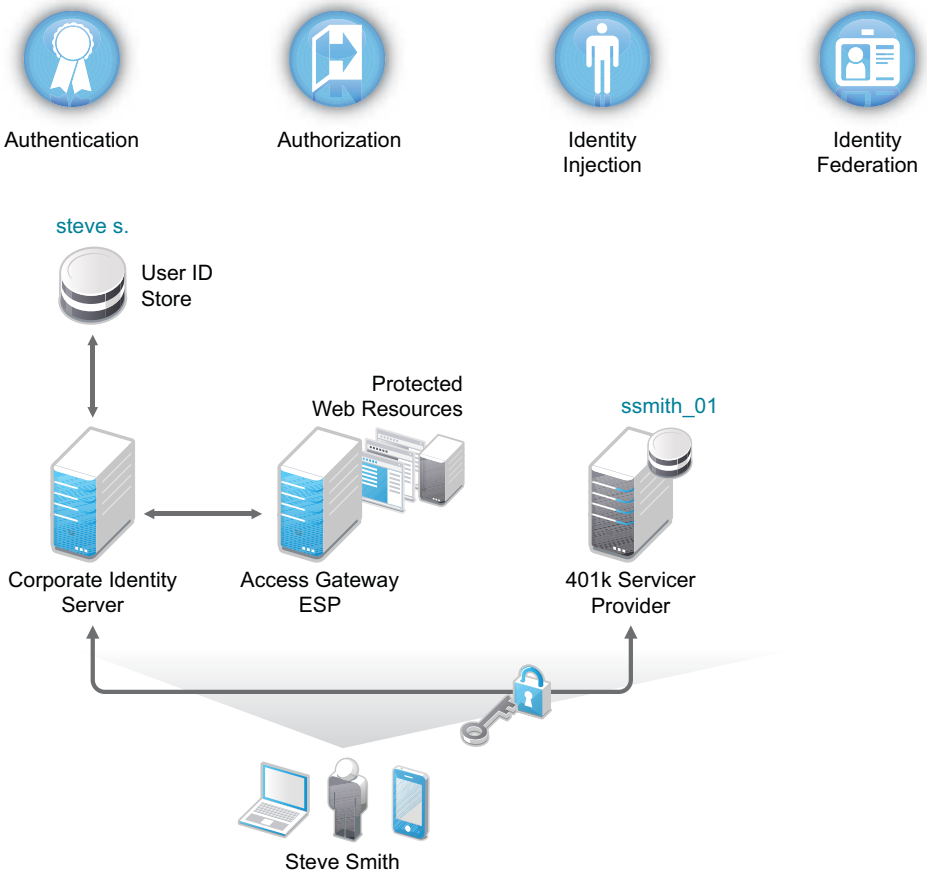
Specifically, Access Manager Appliance helps you stay in compliance with Sarbanes-Oxley, HIPAA, European Union privacy laws, and other regulatory requirements. For an internal assessment or an external auditor, Access Manager Appliance can generate the reports you need, turning compliance requirements into opportunities to develop and implement processes that improve your business practices.

How Access Manager Appliance Works

Access Manager Appliance deployments typically use Identity Servers and Access Gateways to provide policy-driven access control for HTTP services.

Figure 1-1 illustrates the primary purposes of Access Manager Appliance: authentication, identity federation, authorization, and identity injection.

Figure 1-1 *Access Manager Appliance*



Authentication

The Identity Server facilitates authentication for all Access Manager Appliance components. This authentication is shared with internal or external service providers on behalf of the user by means of assertions. Access Manager Appliance supports a number of authentication methods, such as name/password, RADIUS token-based authentication, X.509 digital certificates, Kerberos, Risk-based authentication, Time-Based One-Time Password (TOTP), Social authentication and OpenID Connect. You specify authentication methods in the contracts that you want to make available to the other components of Access Manager Appliance, such as Access Gateway.

User data is stored in user stores. User stores are LDAP directory servers to which end users authenticate. You can configure a user store with more than one replica to provide load balancing and failover capability.

Authorization

Authentication is the process of determining who a user is. Authorization is the process of determining what a user is allowed to do. Access Manager Appliance allows you to configure roles and authorization policies, based on criteria other than authentication, to protect a resource. Authorization policies are dynamically applied after authentication and are enforced when a user attempts to access a protected resource.

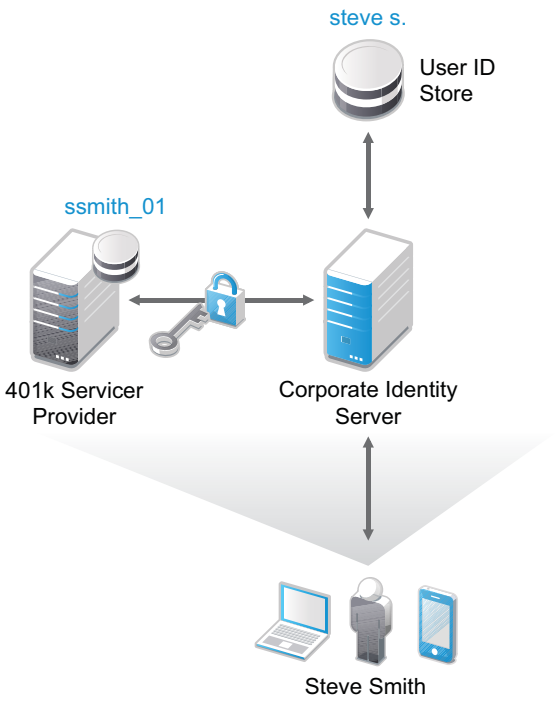
Identity Injection

An Access Gateway lets you retrieve information from your LDAP directory, use it to inject information into HTML headers, query strings, or basic authentication headers, and send this information to the back-end Web servers. Access Manager Appliance calls this technology *identity injection*. The Web server uses this information to personalize content, or can use it for additional authorization decisions. Where Web servers require additional authentication, Identity Injection can also provide the necessary credentials to perform a single sign-on.

Identity Federation

Identity federation is the association of accounts between an identity provider and a service provider. As shown in Figure 1-2, an employee named Steve is known as steve s. at his corporate identity provider. He has an account at a work-related service provider called 401k, which has set up a trust relationship with his company. At 401k he is known as ssmith_01.

Figure 1-2 Identity Federation



As a service provider, 401k can be configured to trust the authentication from the corporate identity provider. Steve can enable single sign-on and single logout by federating or linking his two accounts.

From an administrative perspective, this type of sharing reduces identity management costs, because multiple organizations do not need to independently collect and maintain identity-related data, such as passwords. From the end user's perspective, this results in an enhanced experience by requiring fewer sign-ons.

Access Manager Appliance Components and Their Features

- "Administration Console" on page 30
- "Identity Servers" on page 31
- "Access Gateways" on page 32
- "MobileAccess" on page 33
- "User Portal" on page 33

Administration Console

Administration Console is the central configuration and management tool for the product. It contains a Dashboard option, which allows you to assess the health of all Access Manager Appliance components.

Administration Console allows you to configure and manage each component. It also allows you to manage resources, such as policies, hardware, and certificates, which are used by multiple components.

Identity Servers

Identity Server is the central authentication and identity access point for all other services. It is responsible for authenticating users and distributing role information to facilitate authorization decisions. It also provides the Liberty Alliance Web Service Framework to distribute identity information.

An Identity Server always operates as an identity provider and can optionally be configured to run as an identity consumer (also known as a service provider), by using Liberty, SAML 1.1, SAML 2.0 or OAuth protocols. As an identity provider, Identity Server validates authentications against the supported identity user store. It is the heart of the user's identity federations or account linkage information.

In an Access Manager Appliance configuration, Identity Server is responsible for managing the following tasks:

- **Authentication:** Verifies user identities through various forms of authentication, both local (user supplied) and indirect (supplied by external providers). The identity information can be some characteristic attribute of the user, such as a role, e-mail address, name, or job description. Advanced authentication mechanisms include Time-Based One-Time Password(TOTP), social authentication using external OAuth providers, and risk-based authentication.
- **Identity Stores:** Links to user identities stored in eDirectory, Microsoft Active Directory, or Sun ONE Directory Server.
- **Identity Federation:** Enables user identity federation and provides access to Liberty-enabled services.
- **Account Provisioning:** Enables service provider account provisioning, which automatically creates user accounts during a federation request.
- **Custom Attribute Mapping:** Allows you to define custom attributes by mapping Liberty Alliance keywords to LDAP-accessible data, in addition to the available Liberty Alliance Employee and Person profiles.
- **SAML Assertions:** Processes and generates SAML assertions. Using SAML assertions in each Access Manager Appliance component protects confidential information by removing the need to pass user credentials between the components to handle session management.
- **Single Sign-On and Logout:** Enables users to log in only once to gain access to multiple applications and platforms. Single sign-on and single logout are primary features of Access Manager Appliance and are achieved after the federation and trust model is configured among trusted providers and the components of Access Manager Appliance.
- **Identity Integration:** Provides authentication and identity services to Access Gateways that are configured to protect Web servers. Access Gateway and other Access Manager Appliance components include an embedded service provider that is trusted by Access Manager Appliance Identity Servers.
- **Roles:** Provides RBAC (role-based access control) management. RBAC is used to provide a convenient way to assign a user to a particular job function or set of permissions within an enterprise to control access. Identity Server establishes the active set of roles for a user session each time the user is authenticated. Roles can be assigned to particular subsets of users based on constraints outlined in a role policy. The established roles can then be used in authorization policies to form the basis for granting and restricting access to particular Web resources.

Access Gateways

An Access Gateway provides secure access to existing HTTP-based web servers. It provides security services (authorization, single sign-on, and data encryption) integrated with the identity and policy services of Access Manager Appliance.

Access Gateway is designed to work with Identity Server to enable single sign-on to protected Web services. The following features facilitate single sign-on to Web servers that are configured to enforce authentication or authorization policies:

- **Identity Injection:** Injects the information into HTTP headers that the web server requires.
- **Form Fill:** Automatically fills in the requested form information.

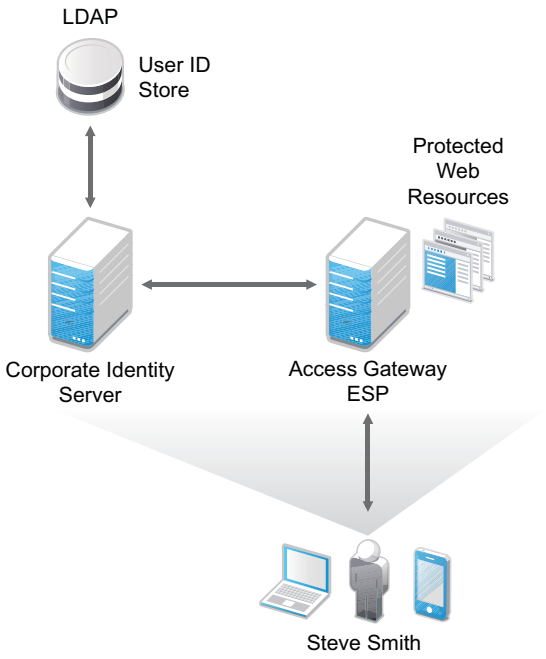
If your Web servers have not been configured to enforce authentication and authorization, you can configure Access Gateway to provide these services. Authentication contracts and authorization policies can be assigned so that they protect the entire Web server or a single page.

Access Gateway can also be configured to cache requested pages. When a user meets the authentication and authorization requirements, the user is sent the page from cache rather than requesting it from the Web server, which enhances the content delivery performance.

Embedded Service Provider

Access Gateway uses an Embedded Service Provider (ESP) to redirect authentication requests to Identity Server. Identity Server allows only trusted devices to participate. To trust each other, devices must exchange metadata. The Embedded Service Provider performs this task automatically for Access Gateway. The logout requests to Identity Server must be digitally signed.

Figure 1-3 *Access Gateway Component*



MobileAccess

Access Manager Appliance allows to manage users' access to resource through mobile devices easily. You create appmarks for each of the Access Manager Appliance resource that you want your users to access. The appmark acts as a bookmark for the resource. You can control access to your resources no matter the type of device or location of the device.

Access Manager Appliance allows you to manage the users' devices and allows users also to manage their own devices. When users lose devices, you can easily remove access to protected resources from those devices through Administration Console. Users can manage their own devices from a dedicated portal page.

The new user portal page displays any appmarks you have created and assigned to the user through the existing Access Manager Appliance roles. The user portal page also presents the different authentication methods assigned to the different resources. Plus you can change easily the branding of the user portal page without editing any JSP files.

User Portal

If you want complex customization of the user portal page, Access Manager Appliance provides a customizable user portal page that is an application where end users can access and manage their authentications, federations, and profile data. The authentication methods you create in Administration Console are reflected in the Portal.

Help information for the end users is provided in the user interface. If you know how to customize JSP pages, you can customize the portal for rebranding purposes and for creating custom login pages.

Language Support

The Access Manager Appliance software for installation and administration uses English and is not localized. Administration Console is also not localized and uses only English. Access Manager Appliance

The User Portal, which appears when the user logs directly into Identity Server, is localized and so is its help file. The User Portal is localized for German, French, Spanish, Italian, Japanese, Portuguese, Dutch, Chinese (Simplified), and Chinese (Traditional). The language must be set in the client's browser to display a language other than English.

Access Gateway and Identity Server, which can send messages to users when an error occurs, allow you to customize the error pages, but you are responsible for supplying the content of the customized pages. For information about customizing these pages, see the following:

- For Access Gateway, see Chapter , "Customizing The Access Gateway," on page 246.
- For Identity Server, see Chapter , "Customizing The Identity Server," on page 192.

Configuring Access Manager

This section describes how to setup a basic Access Manager configuration, perform common administration tasks, and manage components' configuration. For configuring Access Manager, you can use the latest version of Internet Explorer, Chrome, or Firefox browsers.

Topics include:

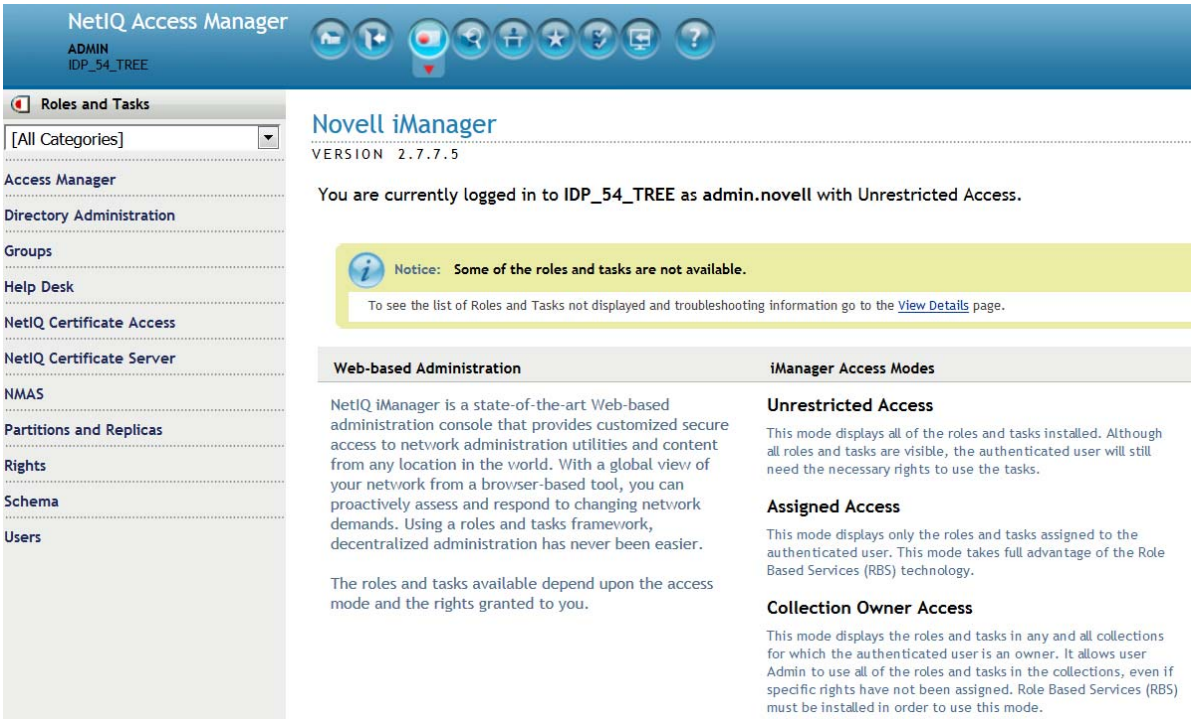
- Chapter 2, "Configuring the Default View," on page 37
- Chapter 3, "Setting Up a Basic Access Manager Appliance Configuration," on page 49
- Chapter 4, "Setting Up an Advanced Access Manager Configuration," on page 189
- Chapter 5, "Configuring Authentication," on page 271
- Chapter 6, "Access Manager Policies," on page 591
- Chapter 7, "Enabling Mobile and Web Access," on page 761
- Chapter 8, "High Availability and Fault Tolerance," on page 771

2 Configuring the Default View

- “Configuring the Default View” on page 37
- “Managing the Administration Console Session Timeout” on page 39
- “Managing Administrators” on page 39
- “Changing the IP Address of Access Manager Appliance” on page 47
- “Changing the DNS Name of Access Manager Appliance” on page 48

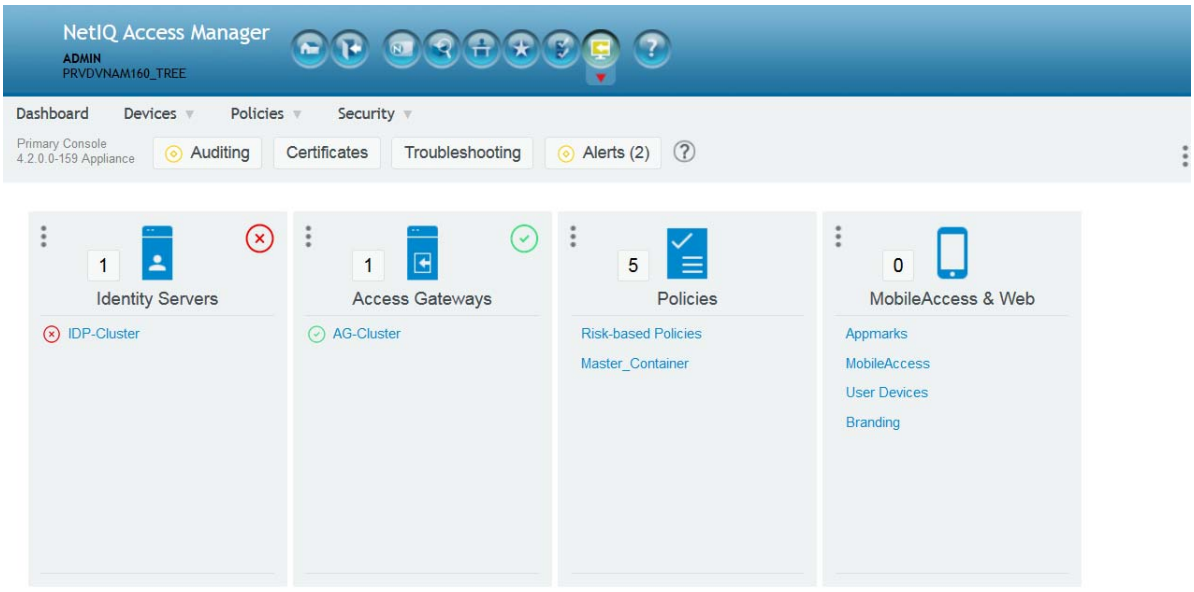
Configuring the Default View

Access Manager Appliance has two views in the Administration Console. Access Manager and its Support Packs used the **Roles and Tasks** view, with Access Manager Appliance the first listed task in the left hand navigation frame. It looks similar to the following:



This view allows you to quickly access other tasks that you occasionally need to manage the configuration of the datastore are visible.

Access Manager Appliance looks similar to the following:



This view has the following advantages:

- You can expand and collapse the containers to see as much or as little as you want. For example, you can see the containers for the policies or you can see the containers and all of the policies in the container.
- You see the Access Manager components in one view. If you expand the view, any item displayed you can click on it and the Administration Console takes you to the configuration page for that component.
- Your common tasks are easily accessed through the dashboard. For example, you quickly access:
  - Auditing
  - Certificates
  - Troubleshooting
  - Alerts
- You can see the health of the different Access Manager components from the dashboard. If the component is healthy, the icons are green. If the components are not healthy, the icons are yellow and red.
- You can navigate faster than in the **Roles and Tasks** view.

When you install or upgrade Access Manager and log in to the Administration Console, the default view is set to the Access Manager view.

Changing the View

1 Locate the Header frame.



2 Click either the Roles and Tasks view  or the Access Manager view  .

Setting a Permanent Default View

1 In the iManager Header frame, click the Preferences view.

2 In the left navigation frame, click **Set Initial View**.

3 Select your preferred view, then click **OK**.

Managing the Administration Console Session Timeout

The `web.xml` file for Tomcat specifies how long an Administration Console session can remain inactive before the session times out and the administrator must authenticate again. The default value is 30 minutes.

To change this value:

1 Change to the Tomcat configuration directory:

`/opt/novell/nam/adminconsole/conf/web.xml`

2 Open the `web.xml` file in a text editor and search for the `<session-timeout>` parameter.

3 Modify the value and save the file.

4 Restart the Administration Console:

`/etc/init.d/novell-ac restart` OR `rcnovell-ac restart`

Managing Administrators

You can create administrators with different access controls manage them in the Administration Console.

The Administration Console notifies you when another administrator makes changes to a policy container or to an Access Manager device such as an Access Gateway. The person who is currently editing the configuration is listed at the top of the page with an option to unlock and with the person's distinguished name and IP address. If you select to unlock, you destroy all changes the other administrator has done.

WARNING: Locking has not been implemented on the pages for modifying the Identity Server. If you have multiple administrators, they need to coordinate with each other so that only one administrator is modifying an Identity Server cluster at any given time.

Multiple Sessions: Do not start multiple sessions of the Administration Console in the same browser on a workstation. Browser sessions share settings that can result in problems when you apply changes to configuration settings. However, if you are using two different brands of browsers simultaneously, such as Internet Explorer and Firefox, it is possible to avoid the session conflicts.

Multiple Administration Consoles: As long as the primary console is running, all configuration changes must be made at the primary console. If you make changes at both a primary console and a secondary console, browser caching can cause you to create an invalid configuration.

The following sections explain how to create additional administrator accounts, how to delegate rights to administrators, and how to manage policy view administrators:

- "Creating Multiple Admin Accounts" on page 40
- "Managing Policy View Administrators" on page 41
- "Managing Delegated Administrators" on page 41

Creating Multiple Admin Accounts

The Administration Console is installed with one admin user account. If you have multiple administrators, you might want to create a user account for each one so that log files reflect the modifications done by each administrator. The easiest way to do this is to create a new user as a trustee of the tree root with [Entry Rights] for Supervisor and inheritable rights assignment This also ensures that you have more than one user who has full access to the Administration Console. If you have only one administrator user and the user forgets the password, you cannot access the Administration Console.

To create a new user as a trustee of the tree root:

1. In Administration Console Dashboard, click icon at the top right of the page and then click the **Roles and Tasks** view in the iManager header.
2. Click **Users** > **Create User.**

   Specify all the required details to create a valid user.

   NOTE: Select the same **Context** that the existing administrator has.
3. Click **Rights > Modify Trustees**, then select the tree root user.
4. Add the newly created user as a trustee of the tree root user.
5. Click **Assigned Rights** and specify [Entry Rights] for supervisor and inheritable rights assignment.
6. Click **Done**.

You can also create delegated administrators and configure them to have rights to specific components of Access Manager. For configuration information for this type of user, see "Managing Delegated Administrators" on page 41.

Managing Policy View Administrators

The super administrators can create policy view administrators. Policy view administrators can log in to Access Manager with their credentials and they can only view the policy containers assigned to them.

The policy view administrators are created same as creating users. For more information on creating users, see “Creating Users” on page 45. In step 5b, select "ou=policyviewusers, o=novell” option in the Context field from the **Contents** list

After creating user, assign rights to the newly created user. For more information, see “Policy Container Administrators” on page 43.

Managing Delegated Administrators

As an Access Manager administrator, you can create delegated administrators to manage the following Access Manager components.

- Individual Access Gateways or an Access Gateway cluster
- Identity Server clusters
- Policy containers

IMPORTANT: You need to trust the users you assign as delegated administrators. They are granted sufficient rights that they can compromise the security of the system. For example if you create delegated administrators with View/Modify rights to policy containers, they have sufficient rights to implement a cross-site scripting attack by using the Deny Message in an Access Gateway Authorization policy.

Delegated administrators are also granted rights to the LDAP server. They can access the configuration datastore with an LDAP browser. Any modifications made with the LDAP browser are not logged by Access Manager. To log LDAP events, you need to turn on eDirectory auditing. For configuration information, see “Activating eDirectory Auditing for LDAP Events” on page 45.

By default, all users except the administrator are assigned no rights to the policy containers and the devices. The administrator has all rights and cannot be configured to have less than all rights. The administrator is the only user who has the rights to delegate rights to other users, and the only user who can modify keystores, create certificates, and import certificates.

The configuration pages for delegated administrators control access to the Access Manager pages. They do not control access to the tasks available for the **Roles and Tasks** view in iManager. If you want your delegated administrators to have rights to any of these tasks such as Directory Administration or Groups, you must use eDirectory methods to grant the user rights to these tasks or enable and configure Role-Based Services in iManager.

To create a delegated administrator, you must first create user accounts, then assign them rights to the Access Manager components.

1. In Administration Console Dashboard, click icon at the top right of the page and then click the Roles and Tasks view.
2. (Optional) If you want to create a container for your delegated administrators, click **Directory Administration** > **Create Object,** then create a container for the administrators.
3. To create the users, click **Users** > **Create User** and create user accounts for your delegated administrators. You can create the users based on the delegatedusers or policyviewusers context. For more information on Creating Users, see “Creating Users” on page 45.

4 Return to the Access Manager view. In Administration Console Dashboard click **Administrators**.

5 Select the component you want to assign a user to manage.

For more information about the types of rights you might want to assign for each component, see the following:

- “Access Gateway Administrators” on page 43
- “Policy Container Administrators” on page 43
- “Delegated Administrators of the Identity Servers” on page 44

6 To assign all delegated administrators the same rights to a component, configure **All Users** option by using the drop-down menu and selecting **None**, **View Only**, or **View/Modify**.

By default, **All Users** is configured for **None**. **All Users** is a quick way to assign everyone View Only rights to a component when you want your delegated administrators to have the rights to view the configuration but not change it.

7 To select one or more users to assign rights, click **Add**, then specify the following details:

Name filter: Specify a string that you want the user’s cn attribute to match. The default value is an asterisk, which matches all cn values.

Search from context: Specify the context you want used for the search. Click the down-arrow to select from a list of available contexts.

Include subcontainers: Specifies whether subcontainers must be searched for users.

8 Click **Query**. The **User** section is populated with the users that match the query.

9 In the **User** section, select one or more users to whom you want to grant the same rights.

10 For the **Access** option, click the down-arrow and select one of the following values:

View/Modify: Grants full configuration rights to the device. View/Modify rights do not grant the rights to manage keystores, to create certificates, or to import certificates from other servers or certificate authorities. View/Modify rights allow the delegated administrator to perform actions such as stop, start, and update the device.

If the assignment is to a policy container, this option grants the rights to create policies of any type and to modify any existing policies in the container

View Only: Grants the rights to view all the configuration options of the device or all rules and conditions of the policies in a container.

None: Prevents the user from seeing the device or the policy container.

11 In the **Device** or **Policy Container**s section, select the devices, the clusters, or policy containers that you want to assign for delegated administration.

12 Click **Apply**.

The rights are immediately assigned to the selected users. If the user already had a rights assignment to the device or policy container, this new assignment overwrites any previous assignments.

13 After assigning a user rights, check the user’s effective rights.

A user’s effective rights and assigned rights do not always match. For example, if Kim is granted View Only rights but All Users have been granted View/Modify rights, Kim’s effective rights are View/Modify.

Access Gateway Administrators

You can assign a user to be a delegated administrator of an Access Gateway cluster or a single Access Gateway that does not belong to a cluster. You cannot assign a user to manage a single member of a cluster.

When a delegated administrator of an Access Gateway cluster is granted View/Modify rights, the administrator has sufficient rights to change the cluster configuration, to stop and start (or reboot and shut down), and to update the Access Gateways in the cluster. However, to configure the Access Gateway to use SSL, you need to be the admin user, rather than a delegated administrator.

When the user is assigned View/Modify rights to manage a cluster or an Access Gateway, the user is automatically granted View Only rights to the master policy container. If you have created other policy containers, these containers are hidden until you grant the delegated administrator rights to them. View Only rights allows the delegated administrator to view the policies and assign them to protected resources. It does not allow them to modify the policies. If you want the delegated administrator to modify or create policies, you need to grant View/Modify rights to a policy container.

View/Modify rights to an Access Gateway or a cluster allows the delegated administrator to modify which Identity Server cluster the Access Gateway uses for authentication. It does not allow delegated administrators to update the Identity Server configuration, which is required whenever the Access Gateway is configured to trust an Identity Server. To update the Identity Server, the delegated administrator needs View/Modify rights to the Identity Server configuration.

Policy Container Administrators

The policy container administrators are of two types:

- Delegated Administrators
- Policy View Administrators

Delegated Administrators

All delegated administrators with View/Modify rights to a device have read rights to the master policy container. To create or modify policies, a delegated administrator needs View/Modify rights to a policy container. When a delegated administrator has View/Modify rights to any policy container, the delegated administrator is also granted enough rights to allow the administrator to select shared secret values, attributes, LDAP groups, and LDAP OUs to policies.

If you want your delegated administrators to have full control over a device and its policies, you might want to create a separate policy container for each delegated administrator or for each device that is managed by a group of delegated administrators.

Policy View Administrators

A policy view administrator has rights only to view policy containers. The super administrators can create a special type of delegated administrators called policy view administrators. The policy view administrators can login to Access Manager with their credentials and they are allowed to view only the policy containers assigned to them.

Using Policy Container option the super administrators can add and remove the delegated and policy view administrators.

- Adding Administrators
- Removing Administrators

Adding Policy Container Administrators

The administrator can assign the rights to the delegated administrators and the users based on the policy containers.

1. In the Administration Console Dashboard, click icon at the top right of the page and then click **Administrators**> **Policy Containers > Add Administrators**.
2. (Optional) Specify the filter.
3. Select the **Access Rights** from the list for the type of administrator. For Example -View/Modify, View Only, and None. The policy view administrators have only **View Only** rights.
4. Select the search from context in the list. For example, “ou=delegated users, o=novell, ou=policyviewusers, o=novell”. Based on the user selected, the delegated or policy view administrators are created.
5. (Optional) Select **Include Subcontainers**, if you want to add it.
6. Click **Query**. The users and the policy containers are displayed for the selected query.
7. Select **User** and **Policy Container**. The users and policy containers list are displayed based on the association with query.
8. Click **Apply** > Close.

Removing Policy Container Administrators

To remove the administrators from the policy containers list, do the following:

1. In the Administration Console Dashboard, click icon at the top right of the page and then click **Administrators**> **Policy Containers** > **Remove Administrators**.
2. Select the check box of the user assigned to the administrator and click **Remove**.
3. Click **Close**.

Delegated Administrators of the Identity Servers

You cannot assign a delegated administrator to an individual Identity Server. You can only assign a delegated administrator to a cluster configuration, which gives the delegated administrator rights to all the cluster members.

When a delegated administrator of an Identity Server cluster is granted the View/Modify rights, the administrator has sufficient rights to change the cluster configuration and to stop, start, and update the Identity Servers in that cluster. The administrator is granted view rights to the keystores for each Identity Server in the cluster. To change any of the certificates, the administrator needs to be the admin user rather than a delegated administrator.

The delegated administrator of an Identity Server cluster is granted View Only rights to the master policy container. If you want the delegated administrator with View/Modify rights to have sufficient rights to manage policies, grant the following rights:

- To have sufficient rights to create Role policies, grant View/Modify rights to a policy container.
- To have sufficient rights to enable Role policies, grant View Only rights to the policy containers with Role policies.

Activating eDirectory Auditing for LDAP Events

If you are concerned that your delegated administrators might use an LDAP browser to access the configuration datastore, you can configure eDirectory to audit events that come from LDAP connections to the LDAP server.

1. In the Administration Console Dashboard, click **Auditing**.
2. Ensure that you have configured the IP address and port to use for your Secure Logging Server.

   The server can be a Novell Audit server, a Sentinel server, or a Sentinel Log Manager. For more information about this process, see "Enabling Auditing" on page 834.

   WARNING: Whenever you change the port or address of the Secure Logging Server, all Access Gateways must be updated. Then every Access Manager device (Identity Server, Administration Console, and Access Gateways) must be rebooted (not just the module stopped and started) before the configuration change takes affect.

   ---
3. From the iManager view bar, select the Roles and Tasks view.
4. Click **Directory Administration** > **Modify Object**.
5. Click the **Object Selector** icon, expand the **novell** container, then select the eDirectory server.

   The eDirectory server uses the tree name, without the _TREE suffix, for its name. The tree name is displayed in the iManager view bar.
6. Click **OK** > **Novell Audit** > **eDirectory**.
7. From the **Meta**, **Objects**, and **Attributes** sections, select the events that you want to monitor for potential security problems.
   - In the **Meta** section, you probably want to monitor changes made to groups and ACLs.
   - In the **Objects** section, you probably want to monitor who is logging in and out and if objects are being created or deleted.
   - In the **Attributes** section, you probably want to monitor when attribute values are added or deleted.
8. Click **Apply**.
9. Restart eDirectory and the Audit Server. Enter the following commands:

   `/etc/init.d/ndsd restart`

   `/etc/init.d/novell-naudit restart` OR `rcnovell-naudit restart`

Creating Users

After creating users, you can assign the role of a delegated administrator or policy view administrator.

1. Log in to Access Manager.
2. Click **Roles and Tasks** > **Users** > **Create User**.
3. **User Name:** Specify the user name. This is a mandatory field.
4. **(Optional) First Name**: Specify the first name of the user.
5. **Last Name**: Specify the name of the delegated administrator user. This is a mandatory field..
6. (Optional) **Full Name**: Specify the full name of the user.

7 **Context:** Specify the context as delegated administrators.This is a mandatory field.

   7a Click object selector icon. The object selector browser displays the Browse and Search tabs.

   7b Click **Browse** tab. Select delegated users option from the **Contents** list. The delegatedusers.novell or policyviewusers.novell is displayed in the context field based on the selection.

8 **Password:** Specify the password and retype the password to confirm it.

NOTE: Failure to enter a password will allow the user to login without a password.

---

9 **(Optional) Simple Password**: Select this check box to set the simple password.

NOTE: Simple Password is not required for normal eDirectory access. The Universal Password feature supersedes Simple Password. When the Universal Password feature is enabled, setting the Simple Password is not required. For more information on the Universal Password feature, refer to Netware 6.5 Documentation (http://www.novell.com/documentation/nw65/?page=/documentation/lg/nw65/universal_password/data/front.html)

---

10 (Optional) Copy from Template or User Object: Copies the attributes from a user template that you've created.

11 (Optional) **Create Home Directory**: You can create a home directory for this new User object if you have sufficient eDirectory rights. To do this, specify the path where you want to create the user's home directory.

   11a Volume: Applies only to NCP-enabled volumes.

   11b Path: You must specify a valid, existing directory path.The last directory typed in the path is the one that is created; all other directories in the path must already exist. For example, if you specify the path corp/home/sclark, the directories corp and home must already exist. The directory sclark is the only directory created.

12 (Optional) Enter or Select the title, location, department, telephone number, fax number, email address of the delegated user from the list.

13 (Optional) Enter the description if there are any to the user. You are able to add, remove and edit the information as per the requirement.

14 Click **OK**.

After creating a user, assign rights to the newly created user. For more information, see “Policy Container Administrators” on page 43.

Changing Administrator’s Password

You can change password of the Administration Console and user store’s administrators.

Changing the Password of the Administration Console Administrator

1 In the Roles and Tasks view, click **Users** > **Modify User**.

2 Click the **Object Selector** icon.

3 Browse to the novell container and select the name of the admin user, then click **OK**.

4 Click **Restrictions** > **Set Password**.

5 Specify a password in **New password** and confirm the password in Retype new password.

6 Click **OK** > OK.

Changing the Administration Password of the User Store Administrator

Perform the following steps to change the admin password of a user store configured for the Identity Server:

1 In the Administration Console, click **Devices** > **Identity Servers** > **IDP-Cluster**.

2 Go to the **Local** tab and click the existing user store name in the user store's list.

3 Enter a password that matches the User Store password in the **Admin password** text box.

4 Confirm the password in the **Confirm password** text box.

5 Click **Apply**.

Changing the IP Address of Access Manager Appliance

NOTE: Changing the primary IP Address of an Access Manager Appliance is not recommended. This may result in corruption of the configuration store. However, you can modify the Listening IP address of Reverse Proxy or the Outbound IP address used to communicate with the Web Server.

To modify the Listening IP Address or Outbound IP address, do the following:

1 In the Administration Console, click **Devices** > **Access Gateways** > Select the device > **New IP** > click **OK**.

2 Add the secondary IP address if applicable to the interfaces from **Network Settings** > **Adapter List**.

3 Configure the DNS from **Network Settings** > **DNS**.

4 Add the Host entries (if any) from **Network Settings** > **Hosts**.

5 Set up the routing (if any) from **Network Settings** > **Gateways**.

6 Under Services, click on **Reverse Proxy/Authentication**. In the Reverse Proxy List, click the proxy service name. Select the newly added cluster member and select the listening IP address for that service.

7 (Optional) If you want to specify the outbound connection to the Web server, click **Web Servers**, then click **TCP Connect Options**. Select the **Cluster Member** and select the IP address from the drop down list against **Make Outbound Connection Using** if you want to select the outbound IP address to communicate with the Web server.

To modify the IP address of the Audit Server:

1 In the Administration Console Dashboard, click **Auditing.**

2 On the Novell Auditing page, change the IP address for the server and, if necessary, the port.

3 Click **OK**.

4 Update all Access Gateways.

5 Reboot all servers, including the Access Gateways, to use the new configuration.

Changing the DNS Name of Access Manager Appliance

To change the DNS name for Access Manager Appliance, modify the DNS settings for Access Gateway. For information about changing the DNS name, see “Viewing and Modifying DNS Settings” on page 244.

3 Setting Up a Basic Access Manager Appliance Configuration

The initial setup consists of installing NetIQ Access Manager Appliance. You must set up the User Stores for Identity Server and configure the Access Gateway to protect resources running on an HTTP Web server.

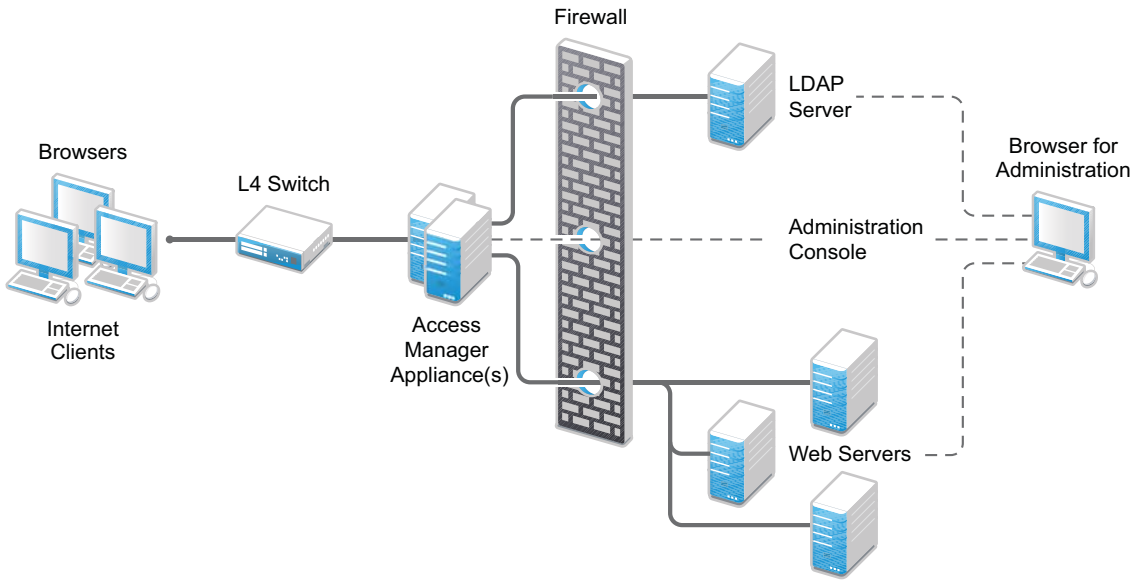
This chapter describes the following topics and tasks:

- “Understanding Access Manager Appliance Process Flow” on page 50
- “Prerequisites for Setup” on page 51
- “Setting up User Stores for Identity Server Configuration” on page 52
- “Identity Servers Cluster” on page 52
- “Configuring the Identity Server Shared Settings” on page 60
- “Configuring the Access Gateway” on page 85
- “Access Gateways Clusters” on page 90
- “Protecting Web Resources Through the Access Gateway” on page 92
- “Configuring Trusted Providers for Single Sign-On” on page 144
- “Configuring Single Sign-On to Specific Applications” on page 174
- “Sample Configuration for Protecting an Application Through Access Manager Appliance” on page 185

Understanding Access Manager Appliance Process Flow

The following figure illustrates the components and process flow that make up a basic configuration.

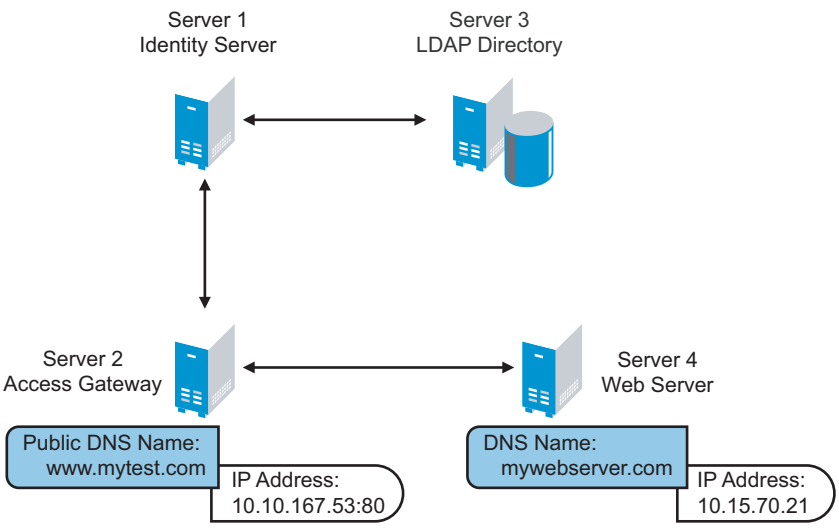
Figure 3-1 *Basic Process Flow*



1. The user sends a request to the Access Gateway for access to a protected resource.
2. The Access Gateway redirects the user to the Identity Server, which prompts the user for a username and password.
3. The Identity Server verifies the username and password against an LDAP directory user store (eDirectory, Active Directory, or Sun ONE).
4. The Identity Server returns an authentication artifact to the Access Gateway through the browser in a query string.
5. The Access Gateway retrieves the user's credentials from the Identity Server through the SOAP channel in the form of a SOAP message.
6. The Access Gateway injects the basic authentication information into the HTTP header.
7. The Web server validates the authentication information and returns the requested Web page.

You configure the Access Manager Appliance so that a user can access a resource on a Web server whose name and address are hidden from the user. This basic configuration sets up communication between the following four servers:

Figure 3-2 *Basic Configuration*



Although other configurations are possible, this section explains the configuration tasks for this basic Access Manager Appliance configuration. This section explains how to set up communication using HTTP. For HTTPS over SSL, see Chapter 15, “Enabling SSL Communication,” on page 819.

Prerequisites for Setup

The following are prerequisites for setting up a basic Access Manager Appliance configuration:

- ❒ Access Manager Appliance is installed.
- ❒ An LDAP directory store with a test user added. This store can be eDirectory, Active Directory, or Sun ONE.
- ❒ A DNS server or modified `host` files to resolve DNS names and provide reverse lookups.
- ❒ A Web server (IIS or Apache). The Web server must have three directories with three HTML pages. The first directory (`public`) must contain a page (such as index.html) for public access. This page needs to provide two links:
  - A link to a page in the `protected` directory. You will configure the Access Gateway to require authentication before allowing access to this page. You do not need to configure the Web server to protect this page.
  - A link to a page in the `basic` directory. You must already have configured your Web server to require basic authentication before allowing access to this page. See your Web Server documentation for instructions on setting up basic authentication. (This type of access is optional, but explained because it is fairly common.)

  If you do not have a Web server that you can use for this type of access, you might prefer to configure Access Manager for the sample Web pages we provide. See Chapter , “Sample Configuration for Protecting an Application Through Access Manager Appliance,” on page 185.
- ❒ A client workstation with a browser with browser pop-ups enabled.

Setting up User Stores for Identity Server Configuration

Post installation, create an Identity Server configuration that defines how an Identity Server or Identity Server cluster operates.

While configuring the user store, specify the following information:

- The IP address of an LDAP directory (user store). The LDAP directory is used to authenticate users. The trusted root certificate of the user store is imported to provide secure communication between the Identity Server and the user store.
- The distinguished name and password of the administrator of the LDAP user store.

1. Configure the User Store. For information about configuring User Store, see “Configuring the User Store” on page 273.
2. (Optional) Verify the configuration:
   - **2a** In a browser, enter the Base URL of Access Manager Appliance. Click the Sample Application Link. You will be redirected to the Login Page.
   - **2b** Log in to using the credentials of a user in the LDAP server.
   - **2c** (Conditional) If the URL returns an error rather than displaying a login page, verify the following:
     - The browser machine can resolve the DNS name of the Identity Server.
     - The browser machine can access the port.

Identity Servers Cluster

After you install Access Manger Appliance, an Identity Server cluster configuration is created automatically. If you install a secondary appliance, the Identity Server in that server will automatically be added to the Identity Server cluster.

In the Access Manager Appliance, Identity Server is automatically configured as a service that is accelerated through the Access Gateway. Access Gateway in one appliance is configured to communicate only to the Identity Server in the same appliance. However, Identity Servers in a cluster can internally communicate to each other through the cluster back channel.

Managing a Cluster of the Identity Servers

Whether you have one machine or multiple machines in a cluster, the Access Manager software configuration process is the same. This section describes the following cluster management tasks:

- “Editing a Cluster Configuration” on page 53
- “Configuring a Cluster with Multiple Identity Servers” on page 55
- “Configuring Session Failover” on page 55
- “Editing Cluster Details” on page 56
- “Enabling and Disabling Protocols” on page 57
- “Configuring Identity Server Global Options” on page 57

Editing a Cluster Configuration

This section discusses all the settings available when editing an Identity Server configuration.

An Identity Server always operates as an identity provider and can optionally be configured to run as an identity consumer (also known as a service provider), by using Liberty, SAML 1.1, SAML 2.0, or WS Federation protocols. These topics are not described in this section.

In an Identity Server configuration, you specify the following information:

- The DNS name for the Identity Server or clustered server site.
- Certificates for the Identity Server.
- Organizational and contact information for the server, which is published in the metadata of the Liberty and SAML protocols.
- The LDAP directories (user stores) used to authenticate users, and the trusted root for secure communication between the Identity Server and the user store.

To edit an Identity Server configuration:

1. In the Administration Console, click **Devices** > **Identity Servers** > **Edit**.
2. Fill in the following fields:

   Name: Specify a name by which you want to refer to the configuration.

   Base URL: Specifies the application path for the Identity Server. The Identity Server protocols rely on this base URL to generate URL endpoints for each protocol. You cannot modify the values in this field.

   - **Protocol:** The communication protocol is HTTPS to run securely (in SSL mode) and for provisioning.
   - **Domain:** Specifies the DNS name assigned to the Identity Server. When you are using an L4 switch, this DNS name must resolve to the virtual IP address set up on the L4 switch for the Identity Servers.
   - **Port:** Default port is 443.
   - **Application:** Specifies the Identity Server application. The default value is nidp.
3. To configure session limits, fill in the following fields:

   LDAP Access: Specify the maximum number of LDAP connections the Identity Server can create to access the configuration store. You can adjust this amount for system performance.

   Default Timeout: Specify the session timeout you want assigned as a default value when you create a contract. This value is also assigned to a session when the Identity Server cannot associate a contract with the authenticated session. During federation, if the authentication request uses a type rather than a contract, the Identity Server cannot always associate a contract with the request.

   Limit User Sessions: Specify whether user sessions are limited. If selected, you can specify the maximum number of concurrent sessions a user is allowed to authenticate.

   If you decide to limit user sessions, you must also give close consideration to the session timeout value (the default is 60 minutes). If the user closes the browser without logging out (or an error causes the browser to close), the session is not cleared until the session timeout expires. If the user session limit is reached and those sessions have not been cleared with a logout, the user cannot log in again until the session timeout expires for one of the sessions.

When you enable this option, it affects performance in a cluster with multiple Identity Servers. When a user is limited to a specific number of sessions, the Identity Servers must check with the other servers before establishing a new session.

- **Deleting Previous User Sessions:** You can configure the Identity Server to delete the previous user sessions if the number of open sessions reaches the maximum limit of allowed sessions that you have specified in the **Limit User Sessions** field. Set the `DELETE OLD SESSIONS OF USER` option to true and restart the Identity Server. For information about how to configure this option, see "Configuring Identity Server Global Options" on page 57. Previous sessions are cleared across Identity Server clusters only when a fresh authentication request comes in. When the Identity server deletes previous user sessions, it sends a logout request to the service provider through the SOAP back channel.

  Use case: In this scenario, a user is accessing a protected resource from a machine and wants to access the same protected resource from another device. The Identity Server will not give access to the user if the **Limit User Sessions** has reached a maximum limit. The Identity Server must terminate the old session of the user so that the user can access the new session seamlessly.

Allow multiple browser session logout: Specify whether a user with more than one session to the server is presented with an option to log out of all sessions. If you do not select this option, only the current session can be logged out. Deselect this option in instances where multiple users log in as guests. Then, when one user logs out, none of the other guests are logged out.

When you enable this option, you must also restart any Embedded Service Providers that use this Identity Server configuration.

4 To configure TCP timeouts, fill in the following fields:

LDAP: Specify how long an LDAP request to the user store can take before timing out.

Proxy: Specify how long a request to another cluster member can take before timing out. When a member of a cluster receives a request from a user who has authenticated with another cluster member, the member sends a request to the authenticating member for information about the user.

Request: Specify how long an HTTP request to another device can take before timing out.

5 To control which protocols can be used for authentication, select one or more of the following protocols:

IMPORTANT: Enable only the protocols that you are using.

If you are using other Access Manager devices such as Access Gateway, you need to enable the Liberty protocol. Access Manager devices use an Embedded Service Provider. If you disable the Liberty protocol, you disable the trusted relationships these devices have with the Identity Server, and authentication fails.

Liberty: Uses a structured version of SAML to exchange authentication and data between trusted identity providers and service providers and provides the framework for user federation.

SAML 1.1: Uses XML for exchanging authentication and data between trusted identity providers and service providers.

SAML 2.0: Uses XML for exchanging encrypted authentication and data between trusted identity providers and service providers and provides the framework for user federation.

WS Federation: Allows disparate security mechanisms to exchange information about identities, attributes, and authentication.

WS-Trust: Allows secure communication and integration between services by using security tokens.

Configuring a Cluster with Multiple Identity Servers

To add capacity and to enable system failover, you can cluster a group of Identity Servers by clustering a group of Access Manager appliances. The Access Manager appliance cluster will automatically cluster the group of Identity Servers. You can also configure the cluster to support session failover, so that users don't have to reauthenticate when an Identity Server goes down.

- ❒ To enable session failover so users don't have to re-authenticate when an Identity Server goes down, see "Configuring Session Failover" on page 55.
- ❒ To modify the name of the cluster or edit communication details, see "Editing Cluster Details" on page 56.

Configuring Session Failover

When you set up an Identity Server cluster and add more than one Identity Server to the cluster, you have set up fault tolerance. When you set up an Identity Server cluster and more than one Identity Servers are added to the cluster, you have set up fault tolerance.This ensures that if one of the Identity Servers goes down, users still have access to your site because the remaining Identity Server can be used for authentication. However, it does not' provide session failover. If a user has authenticated to the failed Identity Server, that user is prompted to authenticate and the session information is lost.

When you enable session failover and an Identity Server goes down, the user's session information is preserved. Another peer server in the cluster re-creates the authoritative session information in the background. The user is not required to log in again and experiences no interruption of services.

Prerequisites

- An Identity Server cluster with two or more Identity Servers.
- Sufficient memory on the Identity Servers to store additional authentication information. When an Identity Server is selected to be a failover peer, the Identity Server stores about 1 KB of session information for each user authenticated on the other machine.
- Sufficient network bandwidth for the increased login traffic. The Identity Server sends the session information to all the Identity Servers that have been selected to be its failover peers.
- All trusted Embedded Services Providers need to be configured to send the attributes used in Form Fill and Identity Injection policies at authentication. If you use any attributes other than the standard credential attributes in your contracts, you also need to send these attributes. To configure the attributes to send, click **Devices** > **Identity Servers** > **Edit** > **Liberty** > **[Name of Service Provider]** > **Attributes**.

Configuring Session Failover

1. In the Administration Console, click **Devices** > **Identity Servers**.
2. In the list of clusters and Identity Servers, click the name of an Identity Server cluster.
3. Click the **IDP Failover Peer Server Count**, then select the number of failover peers you want each Identity Server to have.
   - To disable this feature, select 0.

- To enable this feature, select one or two less than the number of servers in your cluster. For example, if you have 4 servers in your clusters and you want to allow for one server being down for maintenance, select 3 (4-1=3). If you want to allow for the possibility of two servers being down, select 2 (4-2=2).

  If you have eight or more servers in your cluster, the formula 8-2=6 gives each server 6 peers. This is probably more peers than you need for session failover. In a larger cluster, you must probably limit the number of peers to 2 or 3. If you select too many peers, your machines might require more memory to hold the session data and you might slow down your network with the additional traffic for session information.

4 Click **OK**.

How Failover Peers Are Selected

The failover peers for an Identity Server are selected according to their proximity. Access Manager sorts the members of the cluster by their IP addresses and ranks them according to how close their IP addresses are to the server who needs to be assigned failover peers. It selects the closest peers for the assignment. For example, if a cluster member exists on the same subnet, that member is selected to be a failover peer before a peer that exists on a different subnet.

Editing Cluster Details

The Cluster Details page lets you manage the configuration's cluster details, health, alerts, and statistics.

1 In the Administration Console, click **Devices**> **Identity Servers**.

2 Click the name of the cluster configuration.

3 Select from the following actions:

Details: To modify the cluster name or its settings, click **Edit**, then continue with Step 4.

Health: To view the health of the cluster, click the **Health** tab.

Alerts: To view the alerts generated by members of the cluster, click the **Alerts** tab.

Statistics: To view the statistics of the cluster members, click the **Statistics** tab.

4 Modify the following fields as required:

Cluster Communication Backchannel: Specify a communications channel over which the cluster members maintain the integrity of the cluster. For example, this TCP channel is used to detect new cluster members as they join the cluster, and to detect members that leave the cluster. A small percentage of this TCP traffic is used to help cluster members determine which cluster member would best handle a given request. This back channel must not be confused with the IP address/port over which cluster members provide proxy requests to peer cluster members.

- **Port:** Specify the TCP port of the cluster back channel on all of the Identity Servers in the cluster.7901 is the default TCP port.
- **Encrypt:** Encrypts the content of the messages that are sent between cluster members.

NOTE: The Level Four Switch Port Translation feature is not required for the Access Manager Appliance as Identity Server cluster is accelerated through Access Gateway.

- **Port translation is enabled on switch:** Specify whether the port of the L4 switch is different from the port of the cluster member. For example, enable this option when the L4 switch is using port 443 and the Identity Server is using port 8443.
- **Cluster member translated port:** Specify the port of the cluster member.

IDP Failover Peer Server Count: For configuration information, see “Configuring Session Failover” on page 55.

5 Click **OK**, then update the Identity Server when prompted.

Enabling and Disabling Protocols

You can control which protocols can be used for authenticating with an Identity Server configuration. A protocol must be enabled and configured before users can use the protocol for authentication. For tight security, consider disabling the protocols that you are not going to use for authentication.

When you disable a protocol, updating the Identity Server configuration is not enough. You must stop and start the Identity Server.

1 In the Administration Console, click **Devices > Identity Servers** > **Edit**.

2 In the **Enabled Protocols** section, select the protocols to enable

3 To disable a protocol, deselect it.

4 Click **OK**.

5 (Conditional) If you have enabled a protocol, update the Identity Server.

6 (Conditional) If you have disabled a protocol, stop and start the Identity Server.

6a Select the Identity Server, then click **Stop**.

6b When the health turns red, select the Identity Server, then click **Start**.

6c Repeat the process for each Identity Server in the cluster.

Configuring Identity Server Global Options

Global options are applicable for all Identity Servers in a cluster.

NOTE: Access Manager 4.2 onwards, configuring the following options through files is deprecated. You must configure these option by using the Administration Console.

Perform the following steps to configure Identity Server global options:

1 In the Administration Console, click **Devices > Identity Servers > Edit** > **Options**.

2 Click **New**.

3 Set the following properties based on your requirement:

Property Type	Property Value
Allow Auth Policy Execution	Select false to disable the Identity Server to execute authorization policies. The default value is true. For example, see “Executing Authorization Based Roles Policy During SAML 2.0 Service Provider Initiated Request” on page 424.
Cluster Cookie Domain	Set this property to change the Domain attribute for the Identity Server custer cookie. For example, see “Configuring X.509 Authentication to Provide Access Manager Error Message” on page 320.

Property Type	Property Value
Cluster Cookie Path	Set this property to change the Path attribute for the Identity Server custer cookie. The default value is /nidp. For example, see “Configuring X.509 Authentication to Provide Access Manager Error Message” on page 320.
DECODE RELAY STATE PARAM	Select true to enable the relay state URL decoding. The default value is false.
DELETE OLD SESSIONS OF USER	Select true to enable the Identity Server to delete the previous user sessions if the number of open sessions reaches the maximum limit of allowed sessions that you have specified in the **Limit User Sessions** field. The default value is false.
HTTP ONLY CLUSTER COOKIE	Select false to disable the HTTPOnly flags for Identity Server cluster cookies. The default value is true.
IS SAML2 POST INFLATE	Select true to enable the Identity Server to receive deflated SAML 2.0 POST messages from its trusted providers. The default value is false. You can configure post binding to be sent as a compressed option by configuring this property. For example, see the note in Step 4 on page 429.
IS SAML2 POST SIGN RESPONSE	Select true to enable the identity provider to sign the entire SAML 2.0 response for all service providers.
LOGOUT IDP SESSION ON IP CHANGE	Select true to enable Identity Server to logout from the current session if the IP Address changes. The default value is false.
RENAME SESSION ID	Select false to prevent changing the session ID automatically. The default value is true.
SAML1X ATTRIBUTE MATCH BY NAME	Select true to perform a strict check on the name space of the attributes received in assertion. For example, see “SAML 1.1 Service Provider Re-requests for Authentication” on page 995.
SECURE CLUSTER COOKIE	Select false to disable the secure flags for cluster cookies. The default value is true.

Property Type	Property Value
STS CHANGE ISSUER	Specify the value in this format: SPentityID:UPNDomain -> new IssuerID. For example, urn:federation:MicrosoftOnline:support.namnetiq.in -> https://namnetiq.in/nidp/wsfed/ In case of multiple child domains, add each parent domain and child domain separated by a comma. For example, if namnetiq.in is the parent domain and support.namnetiq.in and engineering.namnetiq.in are the child domains, specify the following entries: `urn:federation:MicrosoftOnline:namnetiq.in -> https://namnetiq.in/nidp/wsfed/, urn:federation:MicrosoftOnline:support.namnetiq.in -> https://namnetiq.in/nidp/wsfed/, urn:federation:MicrosoftOnline:engineering.namnetiq.in -> https://namnetiq.com/nidp/wsfed/` For example, see “Configuring Federation for Multiple Domains that Include Child Domains” on page 575.
STS OFFICE365 MULTI DOMAIN SUPPORT AUTO	Select true to enable users to access Office 365 services by using the Issuer URI specific to the domain they belong to. The default value is false. For example, see “Creating Multiple Domains in Office 365 and Establishing Federation with Access Manager” on page 574.
WSF SERVICES LIST	Select full to enable users to access the Services page. Select 404 to return an HTTP 404 status code: Not Found. Select 403 to return an HTTP 403 status code: Forbidden. Select empty to return an empty services list. The default value is full. For example, see “Blocking Access to the WSDL Services Page” on page 371.
WSFED ASSERTION VALIDITY	Specify the assertion validity time in second for WS Federation Provider (SP) to accommodate clock skew between the service provider and SAML identity provider. The default value is 1800 seconds. For example, see “Assertion Validity Window” on page 502.
WSTRUST AUTHORIZATION ALLOWED ACTAS VALUES	Specify the user names who can perform ActAs operations. Allowed user names are the user accounts that the intermediate web service provider uses to authenticate with STS when sending a request with ActAs elements. You can specify more than one user name separated by a comma. For example, see “Adding Policy for ActAs and OnBehalfOf” on page 517.

Property Type	Property Value
WSTRUST AUTHORIZATION ALLOWED ONBEHALF VALUES	Specify the user names who can perform OnBehalfOf operations. Allowed user names are the user accounts that the intermediate web service provider uses to authenticate with STS when sending a request with OnBehalfOf elements. You can specify more than one user name separated by a comma. For example, see “Adding Policy for ActAs and OnBehalfOf” on page 517.
WSTRUST AUTHORIZATION ALLOWED VALUES	Specify the user names who can perform both ActAs and OnBehalfOf operations. You can specify more than one user name separated by a comma. For example, see “Adding Policy for ActAs and OnBehalfOf” on page 517.
OTHER	Specify **Property Name** and **Property Value** if you want to configure any other property.

4 Click **OK** > **Apply**.

Configuring the Identity Server Shared Settings

The **Shared Settings** tab on the Identity Servers page allows you to define the following shared settings:

- **Attribute sets:** Sets of attributes that are exchangeable between identity and service providers.
- **User matching expressions:** The logic of the query to the user store for identification when an assertion is received from an identity provider.
- **Shared Secret names:** Custom shared secret names that you want to be available when configuring policies.
- **LDAP attributes:** Custom LDAP attribute names that you want to be available when configuring policies.
- **Authentication card images:** Custom images that you can assign to authentication cards to uniquely identify an authentication procedure.
- **Metadata Repositories:** Import metadata of trusted providers.

You can reuse these settings. These are available in any Identity Server cluster configuration.

This section describes the following tasks:

- “Configuring Attribute Sets” on page 61
- “Editing Attribute Sets” on page 63
- “Configuring User Matching Expressions” on page 63
- “Adding Custom Attributes” on page 64
- “Adding Authentication Card Images” on page 66
- “Creating an Image Set” on page 67

- “Metadata Repositories” on page 67
- “User Attribute Retrieval and Transformation” on page 68

Configuring Attribute Sets

The attributes you specify on the Identity Server are used in attribute requests and responses, depending on whether you are configuring a service provider (request) or identity provider (response). Attribute sets provide a common naming scheme for the exchange. For example, an attribute set can map an LDAP attribute such as givenName to the equivalent remote name used at the service provider, which might be firstName. These shared attributes can then be used for policy enforcement, user identification, and data injection.

For example, you could have a Web server application that requires the user’s e-mail address. For this scenario, you configure the Web server to be a protected resource of the Access Gateway, and you configure an Identity Injection policy to add the user’s email address to a custom HTTP header. When the user accesses the protected resource, the value of the email attribute is retrieved. However, if you create an attribute set with this attribute, then assign it to be sent with the authentication response of the Embedded Service Provider of the Access Gateway, the value is cached at authentication and is immediately available. If you have multiple attributes that you are using in policies, obtaining the values in one LDAP request at authentication time can reduce the amount of LDAP traffic to your user store.

You can define multiple attribute sets and assign them to different trusted relationships. You can also use the same attribute set for multiple trusted relationships.

To create and configure an attribute set:

1. In the Administration Console, click **Devices > Identity Server > Shared Settings > Attribute Sets > New**.
2. Specify the following fields:

   Set Name: Specify a name for identifying the attribute set.

   Select set to use as template: Select an existing attribute set that you have created, which you can use as a template for the new set, or select **None**. To modify an existing attribute set, select that set as a template.
3. Click **Next**.
4. To add an attribute to the set, click **New**.
5. Specify the following fields:

   Specify the attribute. Select from the following:

   - **Local Attribute:** Select an attribute from the list of all server profile, LDAP, shared secret attributes and virtual attributes. For example, you can select **All Roles** to use in role policies, which enables trusted providers to send role information in authentication assertions. Share secret attributes must be created before they can be added to an attribute set. For instructions, see “Creating Shared Secret Names” on page 64.
   - **Constant:** Specify a value that is constant for all users of this attribute set. The name of the attribute that is associated with this value is specified in the **Remote Attribute** field.

   Remote Attribute: Specify the name of the attribute defined at the external provider. The text for this field is case sensitive.

   - A value is optional if you are mapping a local attribute. If you leave this field blank, the system sends an internal value that is recognized between Identity Servers.

For a SAML 1.1 and SAML 2.0 identity consumer (service provider), a name identifier received in an assertion is automatically given a remote attribute name of *saml:NameIdentifier.* This allows the name identifier to be mapped to a profile attribute that can then be used in policy definitions.

- A value is required if you are mapping a constant.

  An attribute set with a constant is usually set up when the Identity Server is acting as an identity provider for a SAML or Liberty service provider. The name must match the attribute name that the service provider is using.

Remote namespace: Specify the namespace defined for the attribute by the remote system:

- If you are defining an attribute set for LDAP, select **none**. If you want a service provider to accept any namespace specified by an identity provider, select **none**. If you want an identity provider to use a default namespace, select **none**. The `urn:oasis:names:tc:SAML:1.0:assertion` value is sent as the default.
- If you are defining an attribute set for WS Federation, select the radio button next to the text box, then specify the following name in the text box.

  `http://schemas.xmlsoap.org/claims`

- If you want to specify a new namespace, select the radial button by the text box, then specify the name in the text box.

Remote format: Select one of the following formats:

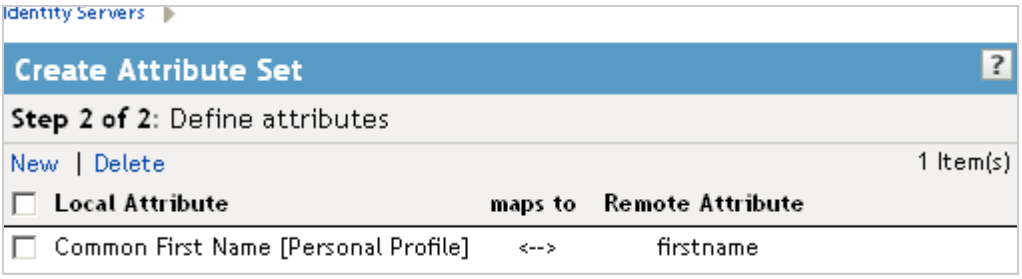
- **unspecified:** Indicates that the interpretation of the content is implementation-specific.
- **uri:** Indicates that the interpretation of the content is application-specific.
- **basic:** Indicates that the content conforms to the xs:Name format as defined for attribute profiles.

Attribute value encoding: Select one of the following encoding options:

- **Special characters encoded:** Encodes only the special characters in the attribute value.
- **Not encoded:** Does not encode the attribute value.
- **Entire value encoded:** Encodes the entire attribute value.

6 Click **OK**.

The system displays the map settings on the Define Attributes page, as shown below:



You can continue adding as many attributes as you need.

7 Click **Finish** after you created the map.

The system displays the map on the Attribute Sets page, as well as indicating whether it is in use by a provider.

8 (Conditional) To configure a provider to use the attribute set, see "Selecting Attributes for a Trusted Provider" on page 155.

Editing Attribute Sets

You can edit attribute sets that have been created in the system.

1. In the Administration Console, click **Devices > Identity Server > Shared Settings > Attribute Sets**.
2. Click the name of the attribute set that you want to edit.
3. The system displays an attribute set page with the following tabs:

   General: Click to edit the name of the attribute set.

   Mapping: Click to edit the attribute map.

   Usage: Displays where the attribute set is used. Informational only.
4. Click **OK**, then click **Close**.

Configuring User Matching Expressions

When a service provider receives an assertion from a trusted identity provider, the service provider tries to identify the user. The service provider can be configured to take one of the following actions:

- Accept that the assertion contains a valid user and authenticate the user locally with a temporary identity and account. As soon as the user logs out, the account and identity are destroyed.
- Use the attributes in the assertion to match a user in the local user store. When you want the service provider to take this action, you need to create a user matching expression.
- Use the attributes in the assertion to match a user in the local user store and when the match fails, create an account (called provisioning) for the user in the local user store of the service provider. When you want the service provider to take this action, you need to create a user matching expression.

The user matching expression is used to format a query to the user store based on attributes received in the assertion from the identity provider. This query must return a match for one user.

- If the query returns a match for multiple users, the request fails and the user is denied access.
- If the query fails to find a match and you have selected provisioning, the service provider uses the attributes to create an account for this user in its user store. If you haven't selected provisioning, the request fails and the user is denied access.

The user matching expression defines the logic of the query. You must know the LDAP attributes that are used to name the users in the user store in order to create the user's distinguished name and uniquely identify the users.

For example, if the service provider user store uses the email attribute to identify users, the identity provider must be configured to send the email attribute. The service provider would use this attribute in a user matching expression to find the user in the user store. If a match is found, the user is granted access. If the user is not found, that attribute can be used to create an account for the user. The assertion must contain all the attributes that the user store requires to create an account.

To create a user matching expression:

1. In the Administration Console, click **Devices > Identity Servers > Shared Settings > User Matching Expressions**.
2. Click **New**, or click the name of an existing user matching expression.
3. Specify a name for the user lookup expression.

4 Click the **Add Attributes** icon (plus sign), then select attributes to add to the logic group. (Use the Shift key to select several attributes.)

5 Click **OK**.

6 To add logic groups, click **New Logic Group**.

The **Type** drop-down (AND or OR) applies only between groups. Attributes within a group are always the opposite of the type selection. For example, if the **Type** value is AND, the attributes within the group are OR.

7 Click the **Add Attributes** icon (plus sign) to add attributes to the next logic group, then click **OK**.

8 Click **Finish**.

9 (Conditional) If you selected attributes from the Custom, Employee, or Personal profile, you need to enable the profile so that the attribute can be shared:

9a Click **Servers** > **Edit** > **Liberty** > **Web Service Provider**.

9b Select the profiles that need to be enabled, then click **Enable**.

9c Click **OK**, then update the Identity Server.

Adding Custom Attributes

You can add custom shared secret names or LDAP attribute names that you want to make available for selection when setting up policies.

- "Creating Shared Secret Names" on page 64
- "Creating LDAP Attribute Names" on page 65

Creating Shared Secret Names

The shared secret consists of a secret name and one or more secret entry names. You can create a secret name only, or a secret name and an entry name. For ease of use, the entry name must match the policy that uses it:

- For a Form Fill policy, the entry name must match a form field name.
- For an Identity Injection policy, the entry name must match the Custom Header Name.
- For an External Attributes policy, **Secret Name** must match the policy name and **Secret Entry Name** must match the attribute name configured while creating the policy.

  For example, if the policy name is fetchattr and attribute name configured in the policy is address, then **Secret Name** must be fetchattr and **Secret Entry Name** must be address.

For more information about how to use shared secrets with policies, see "Creating and Managing Shared Secrets" on page 732.

The Identity Server needs to be configured to use shared secrets. For information about this process, see "Configuring a User Store for Secrets" on page 276.

Shared secret names can be created either on the Custom Attributes page or in the associated policy that consumes them.

1 In the Administration Console, click **Devices > Identity Servers > Shared Settings > Custom Attributes**.

2 To create shared secret names, click **New**.

3 Enter a new shared secret name and, optionally, a secret entry name.

4 Click **OK**.

5 (Optional) To create additional entries for the secret, click the name of the secret, click **New**, specify an entry name, then click **OK**.

WARNING: The Identity Server currently has no mechanism to determine whether a secret is being used by a policy. Before you delete a shared secret, you must ensure that it is not being used.

Creating LDAP Attribute Names

LDAP attributes are available for all policies. LDAP attribute names can be created either on the Custom Attributes page or in the associated policy that consumes them. The attribute names that you specify must match the name of an attribute of the user class in your LDAP user store.

1 In the Administration Console, click **Devices > Identity Servers > Shared Settings > Custom Attributes**.

This list contains the attributes for the inetOrgPerson class. It is customizable.

audio: Uses a u-law encoded sound file that is stored in the directory.

businessCategory: Describes the kind of business performed by an organization.

carLicense: Vehicle license or registration plate.

cn: The X.500 commonName attribute, which contains a name of an object. If the object corresponds to a person, it is typically the person's full name.

departmentNumber: Identifies a department within an organization.

displayName: The preferred name of a person to be used when displaying entries. When displaying an entry, especially within a one-line summary list, it is useful to use this value. Because other attribute types such as cn are multivalued, an additional attribute type is needed.

employeeNumber: Numerically identifies a person within an organization.

employeeType: Identifies the type of employee.

givenName: Identifies the person's name that is not his or her surname or middle name.

homePhone: Identifies a person by home phone.

homePostalAddress: Identifies a person by home address.

initials: Identifies a person by his or her initials. This attribute contains the initials of an individual, but not the surname.

jpegPhoto: Stores one or more images of a person, in JPEG format.

labeledURI: Uniform Resource Identifier with an optional label. The label describes the resource to which the URI points.

mail: A user's e-mail address.

manager: Identifies a person as a manager.

mobile: Specifies a mobile telephone number associated with a person.

o: The name of an organization.

pager: The pager telephone number for an object.

photo: Specifies a photograph for an object.

preferredLanguage: Indicates an individual's preferred written or spoken language.

roomNumber: The room number of an object.

secretary: Specifies the secretary of a person.

sn: The X.500 surname attribute, which contains the family name of a person.

uid: User ID.

userCertificate: An attribute stored and requested in the binary form.

userPKCS12: A format to exchange personal identity information. Use this attribute when information is stored in a directory service.

userSMIMECertificate: PKCS#7 SignedData used to support S/MIME. This value indicates that the content that is signed is ignored by consumers of userSMIMECertificate values.

x500uniqueIdentifier: Distinguishes between objects when a distinguished name has been reused. This is a different attribute type from both the **uid** and the **uniqueIdentifier** type.

2 To add a name:

   2a Click **New**.

   2b If you want the attribute to return raw data rather binary data, select **64-bit Encode Attribute Data**.

   2c Click **OK**.

3 To modify the 64-bit attribute data encoding, click an attribute's check box, then click one of the following links:

   Set Encode: Specifies that LDAP returns a raw format of the attribute rather than binary format, which Access Manager encodes to base64, so that the protected resource understands the attribute. You might use base64 encoding if you use certificates that require raw bites rather than binary string format.

   Clear Encode: Deletes the 64-bit data encoding setting.

4 Click **Apply** to save changes, then click the **Servers** tab to return to the Servers page.

Adding Authentication Card Images

Each authentication contract and managed card template must have a card associated with it.

To add new images, the image files must be available from the workstation where you are authenticated to the Administration Console. Images must fall within the size bounds of 60 pixels wide by 45 pixels high through 200 pixels wide by 150 pixels high.To add a card image:

1 In the Administration Console, click **Devices > Identity Servers > Shared Settings > Authentication Card Images**.

2 Click **New**.

3 Fill in the following fields.

   Name: Specify a name for the image.

   Description: Describe the image and its purpose.

   File: Click **Browse**, locate the image file, then click **Open**.

   Locale: From the drop-down menu, select the language for the card or select **All Locales** if the card can be used with all languages.

4 Click **OK**.

5 If you did not specify **All Locales** for the **Locale,** continue with "Creating an Image Set" on page 67.

Creating an Image Set

You can create card images for specific locales as well as a default image for all locales. The images need to be placed in an image set that allows the browser to display the image associated with the requested locale. If the browser requests a locale for which you have not defined an image, the **All Locales** image is displayed. If an **All Locales** image is not available, the browser displays the **Image not Available** card.

1. In the Administration Console, click **Devices > Identity Servers > Shared Settings > Authentication Card Images**.
2. Click the card image.
3. To add an image to the set, click **New**, then fill in the following fields:

   File: Click **Browse**, then browse to the location of the file.

   Locale: Select the locale from the drop-down menu.
4. Click **OK**.

Metadata Repositories

Large scale federations have more than 100+ identity and or service providers and it is a tedious task to establish bi-lateral relationships with Access Manager. You as an identity provider can now configure several identity and or service providers using a multi-entity metadata file available in a central repository. The identity and/or service providers become partners of a community which maintains a single metadata file containing metadata of all the approved partners. The identity and or service providers submit their metadata which includes specifications of services offered (SAML 1.1 and SAML 2.0) and any other information. This feature is available only for SAML 1.1 and SAML 2.0.

For example, XYZ is an e-book store and several e-book stores, which are either identity or service providers are partner with it. XYZ maintains a single metadata file containing metadata of all the other stores. ABC an e-book identity provider wants to establish a federation with many other e-book stores. Hence, ABC partners with XYZ by sharing its metadata and XYZ in turn shares the metadata XML file. ABC imports the XML file available publicly on the internet (for example, http://xyz.commonfederation.org/xyz-metadata.xml) and establishes trusts with others in the federation which includes XYZ's trusted provider sites.

Creating Metadata Repositories

1. In the Administration Console, click **Devices > Identity Servers > Shared Settings > Metadata Repositories**.
2. Click **New** and fill in the following fields:

   Name: Enter the name of the metadata repository.

   Description: Enter the description of the metadata repository.

   Source: From the drop-down menu, select the source from which you want to import the metadata file.

   - To specify the URL location of the XML file in the **URL** field, select **Metadata URL**.
   - To specify the path of the XML file in the **File** field, select **Metadata File**.
3. Click **Finish**.

   The details of the metadata such as the number of identity servers and service providers present in the metadata, and expiry date of the metadata are displayed.

You can select the metadata repository and click **Delete** to delete the repository. If the metadata file is in use, you cannot delete it. Delete the trusted provider first and then delete the metadata file.

4 Select **All** to see a list of entities. If the entity is supporting it the respective protocol will be checked.

When the metadata repositories are imported, the entities available in the metadata repository can be assigned as trusted provider to any of the Identity Provider clusters. To create trusted providers, see "Managing Trusted Providers" on page 150.

Reimporting Metadata Repositories

You can reimport the metadata repository to get the updated XML.

1 In the Administration Console, click **Devices > Identity Servers > Shared Settings > Metadata Repositories**.

2 Click on the metadata repository you created and click **Reimport**.

3 Specify the URL location of the XML file in the **URL** field and click **Next**.

4 The screen displays the following:

New Entities added to the repositories: If the entities are updated or deleted and are assigned as TrustedProviders to an Identity Server cluster then the Identity Server cluster name is displayed in brackets next to the entity ID.

Entities Deleted from the repositories: If the entity is updated and is assigned as a trusted provider to an Identity Server cluster, that trusted provider will be updated. You must update the Identity Server cluster for the changes to take effect.

Entities Updated in the repositories: If an entity is deleted and was assigned as trusted provider to an Identity Server cluster, then the link between the trusted provider and the metadata repository entity is deleted.

NOTE: The corresponding trusted provider is not deleted and you will have to manually delete the trusted provider.

---

5 Click **Finish** to apply the changes.

User Attribute Retrieval and Transformation

With this release of Access Manager, you can retrieve an attribute from an external source and transform it before sending it in an assertion. The User Attribute Retrieval and Transformation feature enables you to retrieve attributes from a data source (any database or LDAP repositories) and transform them. This feature also allows you to transform user's local attributes (LDAP attributes, Shared Secrets, and various profiles such as Personal Profile and Employee Profile).

Virtual attributes can be used to generate dynamic data at runtime from the current values of the user attributes.

The transformed attribute values are not stored in any persistent data stores. They are in the memory as part of user's session.

- "How User Attribute Retrieval and Transformation Helps" on page 69
- "Prerequisites" on page 69
- "Limitations" on page 69
- "Managing a Data Source" on page 70

- “Managing an Attribute Source” on page 73
- “Managing a Virtual Attribute” on page 75
- “Sample JavaScripts with Examples” on page 79
- “Troubleshooting User Attribute Retrieval and Transformation” on page 85

How User Attribute Retrieval and Transformation Helps

User Attribute Retrieval and Transformation helps you to perform the following activities:

- Retrieve attribute values from external sources other than the configured user stores. The sources can be an external database or any external LDAP repository.
- Transform attribute values before they are sent as part of an assertion to a federated provider. For example, you can edit an attribute value before it is sent from identity provider to a service provider in a SAML 2.0 federation. You can also edit an attribute value sent from identity provider to the Access Gateway used in policies.
- Transform the attribute value used in policies. For example, you can transform the Identity Server role-based policies.

User Attribute Retrieval and Transformation introduces the following configuration settings in the Identity Server:

- Data Source: An entity that holds configuration properties that help in connecting to an external data source. The properties of the data source can be defined in the data source user interface. A data source can be an LDAP repository or a SQL database.
- Attribute Source: Represents queries that fetch attributes from a data source. You can define either an LDAP search filter or a SQL query, based on the data source type.
- Virtual Attribute: Helps you specify the attributes that must be transformed and in the way the transformations happen. A virtual attribute can transform multi- valued attributes.

Prerequisites

To perform complex user attribute transformations, you must have a basic understanding of JavaScript. To see sample JavaScripts with examples, see “Sample JavaScripts with Examples” on page 79.

Limitations

The following are the limitations of the User Attribute Retrieval and Transformation feature:

- Multi-valued input parameters are not supported in the attribute source. If you input a multi-valued parameter, only one value is picked for the calculation.
- You cannot use the existing Identity server user stores directly as an attribute source. You must create a separate data source to use the user stores.
- You cannot edit or store any attribute value permanently in the existing LDAP attributes, shared secret attributes, or external database by using virtual attributes.
- The risk-based policies do not support virtual attributes.
- OAuth configuration does not support virtual attributes.

- Virtual attributes do not support modifications of attribute sources that contain binary data. Only string, boolean, integer, date, and numeric values from database are supported.
- SSL communication with SQL and Oracle databases that are used by virtual attributes (data sources) is not supported.

Managing a Data Source

You can create, edit, or delete a data source.

NOTE: You cannot delete a data source that is being used by an attribute source.

This section discusses the following topics:

- "Creating a Data Source" on page 70
- "Editing a Data Source" on page 72

Creating a Data Source

To create a data source, perform the following steps:

1. In the Administration Console, click **Devices > Identity Server > Shared Settings > Data Sources**.
2. Click **+** to add a data source.
3. From the list, select one of the data sources: **Database** or **LDAP**.
4. Specify the database connection properties.

   4a For Database

   MS SQL and Oracle databases are supported.

Field	Description
Database Name	Specify the name of the database.
Database Driver	Select a driver from the list. The associated driver name is auto-populated. If you select **Others (Unsupported)**, specify the driver name in the adjacent field.
Max Connections	Specify the maximum number of connections. By default, it is set to 20.
Idle TimeOut	Specify the idle timeout. By default, it is set to 600000 milliseconds. You can set this value based on the server setting. For example, if the server timeout is 600000 milliseconds, then the timeout value must not exceed 600000.
Connection TimeOut	Specify the connection timeout. By default, it is set to 10000 milliseconds. You can set this value based on the server setting.
Username	Specify the username used to read from the database.
Password	Specify the password used to read from the database.

Field	Description
Confirm Password	Specify the password once again.
URL	Specify the database URL based on the database driver selected.

Based on the database type you select, you need to add the corresponding jars.

4a1 For Oracle Database:

4a1a Download the JDBC connector for the Oracle database from Oracle.com.

4a1b Copy the JDBC connector jar to the following folder:

On Windows

- Administration Console: `C:\Program Files (x86)\Novell\Tomcat\webapps\nps\WEB-INF\lib`
- Identity Server: `C:\Program Files (x86)\Novell\Tomcat\webapps\nidp\WEB-INF\lib`

On Linux

- Administration Console: `/opt/novell/nam/adminconsole/webapps/nps/WEB-INF/lib`
- Identity Server: `/opt/novell/nam/idp/webapps/nidp/WEB-INF/lib`

4a1c Restart the Administration Console and the Identity Server.

4a2 For Microsoft SQL Server:

4a2a Download the JDBC connector for the SQL Server database from Microsoft.com.

4a2b Copy the JDBC connector jar file to the following folder:

On Windows

- Administration Console: `C:\Program Files (x86)\Novell\Tomcat\webapps\nps\WEB-INF\lib`
- Identity Server: `C:\Program Files (x86)\Novell\Tomcat\webapps\nidp\WEB-INF\lib`

On Linux

- Administration Console: `/opt/novell/nam/adminconsole/webapps/nps/WEB-INF/lib`
- Identity Provider: `/opt/novell/nam/idp/webapps/nidp/WEB-INF/lib`

4a2c Restart the Administration Console and the Identity Server.

4b For LDAP:

eDirectory and Active Directory are supported.

NOTE: The SunONE directory is currently not supported.

Field	Description
LDAP Name	Specify a display name for the LDAP database.
Username	Specify the username used to read from the database.

Field	Description
Password	Specify the password used to read from the database.
Confirm Password	Specify the password once again.
Directory Type	From the list, select the type of directory. If you select **Others (Unsupported)**, specify a directory name in the adjacent field: sunonedir, custom1, custom2, custom3, custom4, others.
Max Connections	Specify the maximum number of connections. By default, it is set to 20.
LDAP Operation TimeOut	Specify the LDAP operation timeout. By default, it is set to 15000 milliseconds. You can set this value based on the server setting.
Idle Connection TimeOut	Specify the connection timeout. By default, it is set to 10000 milliseconds. You can set this value based on the server setting. For example, if the server timeout is 15000 milliseconds, then the LDAP timeout value must not exceed 15000.
IP Address	Specify the IP address of the LDAP directory.
Port	Specify the port number. By default, it is 389. For a secure connection, select **Use Secure LDAP Connection**. The port number changes to 636. You must import the trusted root if you select a secure connection. To import the trusted root, click **Auto Import Trusted Root.** The trusted certificate of the server will be imported to the Identity provider trust store. Update the Identity provider each time.
Search Contexts	The search context is used to locate users in the directory. If a user exists outside of the specified search context and its scope (or One level, Object or Subtree), the Identity Server cannot find the user and the search fails.

5 **Test Connectivity**: Click **Test** to test the data source connection after specifying the details. You can also view the error logs at the following locations:

- Linux: `/opt/novell/nam/logs/adminconsole/tomcat/catalina.out`
- Windows: `\Program Files (x86)\Novell\Tomcat\logs\stdout.log`

Editing a Data Source

1 In the Administration Console, click **Devices > Identity Server > Shared Settings > Data Sources**.

2 Click the data source you want to modify.

3 On the **Edit Data Source** page, modify the details as required.

NOTE: If you change the IP address of the LDAP data source, then, you must import the trusted root of the updated server to the Identity server trust store.

For more information about the fields on this page, see “Creating a Data Source” on page 70.

4 Click **OK**.

5 Update the Identity Server.

IMPORTANT: You must update the Identity Server every time you edit the properties of a data source that is being used by an attribute source and the attribute source in turn, being used by the virtual attribute.

Managing an Attribute Source

You can create, edit, or delete an attribute source.

NOTE: You cannot delete an attribute source that is being used by a virtual attribute.

This section discusses the following topics:

- “Creating an Attribute Source” on page 73
- “Editing an Attribute Source” on page 75

Creating an Attribute Source

1 In the Administration Console, click **Devices > Identity Server > Shared Settings > Virtual Attributes > Attribute Source**.

2 Click **+** to add an attribute source.

3 Specify an attribute source Name.

4 Specify the description of the attribute source.

5 From the list, select the data source.

6 Click **Step 1: Provide input parameters**.

   6a **Name**: Specify a name for the attribute. You must specify the same name in the query field. The default value is %P1%.

   6b **Parameter Value**: Select an attribute from the list. The available options are: Server profile, LDAP, and shared secret attributes.

   6c Click **Show / Add Test Values?** to display the test value field. Specify a value in the Test value field.

   To provide additional parameter values, click **+**.

   NOTE: The attribute source does not support multi-valued inputs. If you input multiple values, only one value is picked for the calculation.

7 Click **Step 2: Provide query and output parameters**.

For one attribute source, you can provide only one output parameter.

7a **Filter / Query**: Specify an LDAP filter or a database query. The query must use the input details specified in the **Step 1: Provide input parameters** section.

7b **Filter / Query Output Parameters**: Specify a name for the filter / query output. To add multiple output parameters, click **Add**.

- In the case of an LDAP filter, you need to specify the exact name of the attribute that you want to fetch.
- In the case of a database query, you can specify an alias for the attribute fetched. The order of the output parameters must match the sequence in which they are specified in the database query.

7c **Test**: To test the input values based on the filter and output parameters, click **Test**. For security reasons, you are prompted to enter the data source credentials. The **Test Result** window displays the status along with the test results. You can also view the error logs at the following locations:

- Linux: `/opt/novell/nam/logs/adminconsole/tomcat/catalina.out`
- Windows: `\Program Files (x86)\Novell\Tomcat\logs\stdout.log`

LDAP Scenario:

Consider a scenario where you want to fetch an email address from the external LDAP directory for which the user's LDAP attribute (from the external LDAP directory) UID matches with the local LDAP attribute cn.

To configure this, perform the following steps:

1. In **Step 1: Provide input parameters**, select LDAP attribute: `cn` as a parameter value. Add input parameter %P1% and map it to the LDAP attribute.
2. In **Step 2: Provide filter and output parameters**:
   a. Specify the following query: `(&(objectclass=*)(uid=%P1%))`
   b. Specify the filter output name as `email`. `email` is the alias name given for the column email.
3. To test this configuration:
   a. In **Step 1: Provide input parameters**, select **Show / Add Test Values?** and provide the test value as `admin`.
   b. In **Step 2: Provide filter and output parameters**: Click **Test**. Enter the data source credentials.

The test results return the email address stored in the directory: `admin123@example.com`.

Database Scenario:

Consider a scenario where you want to fetch an email address from the database for which the user's name matches with the local LDAP attribute cn.

1. In **Step 1: Provide input parameters**, select LDAP attribute: `cn` as a parameter value.
2. In **Step 2: Provide query and output parameters**:
   a. Specify the following query: `select email from Emp where name = '%P1%'` (email and name is the column name and Emp is the table name)
   b. Specify the filter output name as `mail` (mail is the alias name given for the column email)

3. To test this configuration:
    a. In **Step 1: Provide input parameters**, select **Show / Add Test Values?** and provide a test value that represents a record in the column of the table.
    b. In **Step 2: Provide query and output parameters**: Click **Test**. Enter the data source credentials.

The test results return the email address stored in the database: `admin123@example.com`.

Editing an Attribute Source

To edit the properties of an attribute source, perform the following steps:

1 In the Administration Console, click **Devices > Identity Server > Shared Settings > Virtual Attributes > Attribute Source**.

2 Click the attribute source you want to modify.

3 On the **Edit Attribute Source** page, modify the details as required.

For more information about the fields on this page, see “Creating an Attribute Source” on page 73.

4 Click **OK**. Update the Identity Server.

IMPORTANT: If the attribute source is being used by a virtual attribute, you need to update the Identity Server every time you edit the properties of an attribute source.

Managing a Virtual Attribute

You can create, edit, or delete a virtual attribute.

NOTE: You cannot delete a virtual attribute that is being used by an attribute set. Also, before deleting a virtual attribute, ensure that it is not being used by a policy.

This section discusses the following topics:

- “Creating a Virtual Attribute” on page 75
- “Editing a Virtual Attribute” on page 79

Creating a Virtual Attribute

You can create a virtual attribute from an attribute source and an external data source.

1 In the Administration Console, click **Devices > Identity Server > Shared Settings > Virtual Attributes > Virtual Attribute**.

2 Click **+** to create a virtual attribute.

3 Specify a name for the virtual attribute.

4 Enter the description.

5 Click **Step 1: Provide input > parameters.**

5a **Name**: Specify a name for the attribute. If you use advanced JavaScript option, then, you need to specify the same name in the advanced JavaScript field. The default value is P1.

5b **Parameter Value**: Select an attribute from the list. In addition to Server profile, LDAP, and shared secret attributes, attribute sources are also listed. To specify additional values, click **+**.

NOTE: If an attribute source returns a null or an empty value, then the corresponding input parameter takes an empty string value.

5c Click **Show / Add Test Values?** to display the test value field. Click **+** to add a test value. Click the edit icon to edit the test value. Click the delete icon to delete the test value.

6 Click **Step 2: Provide a modification function**.

6a **Select a function**: From the list, select a function. The corresponding JavaScript is displayed in the **Script** field. To view it, expand the script. You can further customize these scripts and use them in the **Advanced JavaScript** field.

The following table lists the pre-defined JavaScript functions along with examples:

Function	Description	Example: Pre-Defined Functions
To UpperCase	Converts the input value to upper case.This function works on arrays and single-valued input. It uses the following JavaScript function: `toUpperCase()` Works only on one input parameter that is selected in **Step 1: Provide input parameters**	If P1=alice, then the result displays ALICE.
To LowerCase	Converts the input value to lower case.This function works on arrays and single-valued input. It uses the following JavaScript function: `toLowerCase()` Works only on one input parameter that is selected in **Step 1: Provide input parameters**	If P1=ALICE, then the result displays alice.
Remove Substring	Removes a substring from all instances of the input value. This function does not remove a substring from the global option.This function works on arrays and single-valued input. It uses the following JavaScript function: `split()` and `join()` Works only on one input parameter that is selected in **Step 1: Provide input parameters**	If P1=a@microfocus.com Remove=@microfocus, then the result is a.com.

Function	Description	Example: Pre-Defined Functions
Find and Replace	Finds and replaces a string from all instances of the input value. Works only on one input parameter that is selected in **Step 1: Provide input parameters**	If P1=abcde Find=e Replace=a, then the result displays abcda
Regex Replace	Finds and replaces a substring from all instances of the input value by using a regular expression. For example, to search /, you must escape it first using \. Use the following syntax: `/\//` This function works on arrays and single-valued input. It uses the following JavaScript functions: `replace()` Works only on one input parameter that is selected in **Step 1: Provide input parameters**	If P1=bob@novell.com Find=@novell.com Replace=@microfocus.com The result displays: bob@microfocus.com
Find Subset by Regex	Use this function if an input is multi-valued and you want a subset of values from it, satisfying a particular condition by using a regular expression.This function works on arrays and single-valued input. It uses the following JavaScript function: `replace()` Works only on one input parameter that is selected in **Step 1: Provide input parameters**	If P1="a@novell.com, b@novell.com,c @microfocus.com, d@microfocushr.com" regex= /microfocus/ Then, the result displays: c@microfocus.com, d@microfocushr.com
Concatenate Values in a Parameter	Concatenates multiple values of a multi-valued input. You must add a separator between the values that you want to concatenate Works only on one input parameter that is selected in **Step 1: Provide input parameters**	If P1=abc, def Separator=+ Then, the result displays: abc+ def

Function	Description	Example: Pre-Defined Functions
Concatenate Multiple Parameters	Concatenates multiple input parameter values, where each input parameter can be multi-valued input. You must add a separator between the values that you want to concatenate	If P1=abc, def and P2=123, 456 Parameter Separator=+ Multi value Separator=: Then, the result displays abc:def+123:456
Advanced JavaScript	Specify a customized JavaScript In this field. You need to create a JavaScript function with name “main” and specify the code in it. You can write your custom code or you can also copy the existing pre-defined code.You can also call multiple functions in the “main” function.	See the “Sample JavaScripts with Examples” on page 79.

NOTE

- After JavaScript processing, if the output is a null value, then the value of the virtual attribute is empty.
- The pre-defined function can handle both single-valued and multi-valued inputs. If the input is multi-valued, then pre-defined function is applied on each of the values.

Advanced JavaScript:

Sample JavaScript:

```
function main(P1, P2)
     {
          //some logic
          //you can call yourFunction(name) and use its return value
          return some value;
     }
     function yourFunction(name)
     {
          //some code
         //return some value;
     }
```

In case of advanced JavaScript, the input parameter name in the main function of the JavaScript must match the input parameter name specified in the **Step 1: Provide input parameters** section. The return value can be a single value or an array.

When the input is multi-valued, it is passed as an array to the main function.

When the Identity Server computes the value of a virtual attribute, it calls a function named “main” that is available in the script provided for it. The value (single value or array) returned by main is the value of the virtual attribute.

For example:Consider a scenario where P1 contains bmw and nissan, you can use the JavaScript `instanceof` function to check if the input is single-valued or multi-valued. If it is multi-valued, then JavaScript iterates over the values `P1=['bmw', 'nissan']`

```
function main (P1){
  if( P1 instanceof Array) {
    var a =P1[0]    //will assign 'bmw' value to variable a
    //do something
  }
}
```

The following code checks if an input parameter has a value, is empty or undefined:

```
function main(P1){
  if(hasNoValue(P1))
    // do something
       return something;
}
function hasNoValue(P1){
   if(P1 == null || (typeof P1 == 'undefined') || P1.trim().length == 0)
        return true;
   else
        return false;
}
```

6b **Test**: To test the input values based on the modification function, click **Test**. To test multi-valued inputs, click the + button.

For example, if an attribute `mail` has two values: `abc@example.com` and `def@example.com`, then to test them, click the + button twice. In each field, add the values separately.

The Test Result window displays the status along with the test results. You can view the error logs at the following locations:

- Linux: `/opt/novell/nam/logs/adminconsole/tomcat/catalina.out`
- Windows: `\Program Files (x86)\Novell\Tomcat\logs\stdout.log`

7 Click **OK**.

8 Update the Identity Server.

Editing a Virtual Attribute

1 In the Administration Console, click **Devices > Identity Server > Shared Settings > Virtual Attributes**.

2 Click the virtual attribute you want to modify.

3 On the **Edit Virtual Attribute** page, modify the details as required.

For more information about the fields on this page, see “Creating a Virtual Attribute” on page 75.

4 Click **OK**. Update the Identity Server.

IMPORTANT: You must update the Identity Server every time you edit the properties of a virtual attribute.

Sample JavaScripts with Examples

The following section provides sample JavaScripts with examples. These are used in the Virtual Attributes section.

Use Case 1:

Consider a scenario where a service provider wants to append PID with an attribute partnerId. For example: PID: P1.

To achieve this, fetch a user's partnerId by using their existing "givenName" LDAP attribute (available from the logged in user store) from the external LDAP repository. Now, and add a string "PID:" to it. Later, send the value in Web servers through the Identity injection policy.

Solution: The solution is as follow:

Creating a Data Source:

1. Configure an LDAP data source with name "`dsLdap`". Specify the connection properties. Test the connection.
2. Import the secure LDAP certificate to Identity Server trust store using the create Data Source screen.
3. Click **Update All** to update the Identity Servers.

Creating an Attribute Source:

1. In the Administration Console, click **Devices > Identity Server > Shared Settings > Virtual Attributes > Attribute Source**. Create an attribute source with name "`dsLdapAttrSrc`".
2. Select data source name "`dsLdap`".
3. Add input parameter `%P1%`. Map it to the LDAP attribute: `givenName`.
4. Add a Filter: name=`%P1%`.
5. Add output parameter: `partnerID`
6. Test Filter: Test the input values.

Creating a Virtual Attribute:

1. In the Administration Console, click **Devices > Identity Server > Shared Settings > Virtual Attributes > Virtual Attribute**. Create a virtual attribute with name "`partnerID`".
2. Add input parameter P1. Map it to `dsLdapAttrSrc:partnerID` (the attribute source that you created in Step 1 of the creating an Attribute Source section).
3. In **Step 2: Provide query and output parameters**, specify the following script:

```
function main(P1){
   return "PID:"+P1;
}
```

4. Test the script. The results return: PID: P1. For example, if partnerID=part123, then, the test result is PID:part123.
5. Update the Identity Server.
6. Use it in the Identity injection policy.

Use Case 2:

Consider a scenario where, if the authenticated user is a manager and has administrator rights to a protected human resource application, then, when he accesses this application, his roles must be passed to the application.

In this scenario, the manager has a local LDAP attribute `isManager` and has roles of a recruiter and an employee. He also has a local LDAP attribute `groupmembership`, which contains the string "admins" (for example: adminsRecruitmentDep, adminsPoliciesDep).

Solution:

Create a virtual attribute: `App1Admin`.

In **Step1: Provide input parameters**, select the following input parameters:

- P1: is mapped to LDAP attribute isManager
- P2: is mapped to LDAP attribute groupmembership
- P3: is mapped to LDAP attribute role value

Use the following code in the **Step 2: Provide a modification function > Advanced Javascript**:

```
function main(P1, P2, P3){
   if(P1 == 'true' && (/admins/i.test(P2) == true)){
       return P3;
   }else{
       return 'NA';
   }
}
```

To test the JavaScript, click the + and add multiple test values. Specify the following test values:

- P1: true
- P2: adminsRecruitmentDep
- P3: recruiter,employee

Output: The output is a multi-valued virtual attribute `recruiter,employee.`

Explanation: In the function, /admins/i.test(P2) == true, /admins/i is a regular expression and test is a JavaScript in-built function. This function tests the regular expression in the string passed as the input parameter. The function returns true if the string contains the required pattern.

Use Case 3:

Consider a scenario where, an Access Manager user wants to access Amazon Web Services (AWS). AWS has multiple roles and each AWS role can have various access rights or policies assigned to it. Based on the level of access, you can access authorized Amazon services. This information about roles must be passed dynamically by Access Manager to AWS to provide single sign-on to the Access Manager user.

For more information about AWS configuration, see “Integrating Amazon Web Services with Access Manager” on page 567.

In this scenario, you have a constant value created using <Role ARN, Trusted SAML Provider ARN> mapped to Remote AWS attribute Role (this value is the AWS format).

Let suppose you have configured the following roles in AWS: admin, finance. The role ARNs are:

- For admin: `arn:aws:iam::638116851885:role/admin`
- For finance: `arn:aws:iam::638116851885:role/finance`

For admin role, pass the following: `arn:aws:iam::638116851885:role/admin,arn:aws:iam::638116851885:saml-provider/NAMIDP`

For finance role, pass the following: `arn:aws:iam::638116851885:role/finance,arn:aws:iam::638116851885:saml-provider/NAMIDP`

In this example, to dynamically generate the AWS role, use the LDAP attribute `Department Name` in the user store. For the admin user, the department name is admin. For the finance user, the department name is finance. To make department name available as an LDAP attribute, ensure that you enable personal profile. Click **Identity Servers > Edit > Liberty > Web Service Provider**.

Solution:

Create a virtual attribute with the following information:

When the user logs in, the department name (finance) is fetched for the respective user and appended with the constant value of the role ARN. This value is then concatenated with the trusted SAML provider ARN in the following format: `arn:aws:iam::638116851885:role/admin,arn:aws:iam::638116851885:saml-provider/NAMIDP`

Map this virtual attribute with the AWS Remote Attribute role.

In **Step1: Provide input parameters**, select P1 parameter value as Department Name (Personal Profile).

Use the following code in the **Step 2: Provide a modification function > Advanced Javascript**:

```
function main(P1){
   var role_arn='arn:aws:iam::638116851885:role/'
   var provider_arn=',arn:aws:iam::638116851885:saml-provider/MyIDP_184-142';
   var aws_role;
   aws_role = role_arn+P1+provider_arn;
   return aws_role;
}
```

To test the JavaScript, click the + and add multiple test values. Specify the test value of P1: finance.

Output: `arn:aws:iam::638116851885:role/finance,arn:aws:iam::638116851885:saml-provider/NAMIDP.`

Use Case 4:

Consider a scenario where, for a service provider - cloudsp, you want to send the groups associated with the user to the service provider. However, you only want to send the groups relevant to that service, and not the complete group DN. Check for a function that checks if the group cn starts with “cloudsp”. If available, send it to the group cn.

In this scenario, the cn of the groups relevant to cloudsp start with “cloudsp” (for example: "cn=cloudspa,ou=group,o=netiq") So, when a cloudsp user authenticates at the Identity provider, you need to extract all the cn values from the local LDAP attribute `groupMembership` and filter only those names starting with cloudsp and send it in assertion to cloudsp.

Solution:

In **Step1: Provide input parameters**, select P1 as an attribute which has the groups.

Use the following code in the **Step 2: Provide a modification function > Advanced Javascript**:

```
function main( P1 ){
    return mapGroups(P1);
}

function mapGroups(attribute){
    var result = [];
    if(attribute instanceof Array){
        var j =0;
        for(var i=0; i<attribute.length; i++){
            var grp = checkGroup(attribute[i]);
            if( grp != 'NA')
               result[j++] = grp;
         }
    }else{
        var grp = checkGroup(attribute);
```

```
        if( grp != 'NA')
            result[0] = grp;
    }
    return result;
}

function checkGroup(group){
    if(/^cn=cloudsp.*,/.test(group) == true){
        var startindex = 3;// it starts with cn
        var endindex = group.indexOf(",");
        return group.substring( startindex, endindex);
     }else
        return 'NA';
}
```

To test the JavaScript, click the + and add multiple test values. Specify the test values:

```
cn=cloudspgroupa,ou=group,o=netiq cn=cloudspgroupb,ou=group,o=netiq
cn=cloudspgroupk,ou=group,o=netiq

 cn=testgroupa,ou=group,o=netiq
```

Output:

```
cloudspgroupa
cloudspgroupb
cloudspgroupk
```

Explanation:

The JavaScript in-built string function substring is used to extract the cn value from the group./ ^cn=cloudsp.*,/.test(group) is a regular expression which matches a string that starts with cloudsp. It has 0 or more characters followed by a comma (,)

Use Case 5: (utility function reuse test case)Consider a scenario where, the Identity provider roles are in the format `companyX:rolename`, but a service provider `abc` wants the roles in the following format: `rolename` and is in upper case.

To achieve this, remove 'companyX:' from each role and convert each of them into upper case for sending them to the protected Web server. Each role is specified as companyX:rolename.

For example: companyX:admin, companyX:guest.

Solution:

In **Step 1: Provide input parameters**, select P1: All Roles.

Use the following code in the **Step 2: Provide a modification function > Advanced Javascript**:

Copy the JavaScript from the following pre-defined functions: Remove Substring and To upperCase.

Remove Substring function:

```
function findReplace(attribute, findString, replaceString){
    var result;
    if(attribute instanceof Array){
         result = [];
         for(var i=0; i<attribute.length; i++){
             result[i] =attribute[i].split(findString).join(replaceString);
         }
    }else{
         result = attribute.split(findString).join(replaceString);
    }
    return result;
}
```

To upperCase function:

```
function convertToUpperCase (attribute){
    var result ;
    if(attribute instanceof Array){
        result = [];
        for(var i=0; i<attribute.length; i++)
            result[i] = attribute[i].toUpperCase();
    }else{
        result = attribute.toUpperCase();
    }
    return result;
}
```

Now, customize the code. In the **Substring to remove** parameter for findReplace (), specify companyX:

```
function main(P1){
    return convertToUpperCase(findReplace (P1, 'CompanyX:'));
}

function findReplace(attribute, findString, replaceString){
    var result ;
    if(attribute instanceof Array){
        result = [];
        for(var i=0; i<attribute.length; i++){
            result[i] =attribute[i].split(findString).join(replaceString);
        }
    }else{
        result = attribute.split(findString).join(replaceString);
    }
    return result;
}

function convertToUpperCase (attribute){
    var result;
    if(attribute instanceof Array){
        result = [];
        for(var i=0; i<attribute.length; i++)
            result[i] = attribute[i].toUpperCase();
    }else{
        result = attribute.toUpperCase();
    }
    return result;
}
```

To test the JavaScript, add the following test values in P1: 'companyX:admin', 'companyX:guest'.Output: `ADMIN, GUEST`.

Use Case 6:

Consider a scenario where you do not want to modify an attribute value that is retrieved from an external source. To send the same attribute value in the assertion to a federated provider or in the policies, perform the following:

In the Administration Console, click **Devices > Identity Server > Shared Settings > Virtual Attributes > Virtual Attribute**.

In **Step1: Provide input parameters**, select P1 and map it to an attribute retrieved from an external source.

In **Step 2: Provide a modification function**, select **Advanced JavaScript** and specify the following script:

```
function main(P1){
    return P1;
}
```

Test the script. The results returns the value of the attribute source specified as P1.

Update the Identity Server.

Troubleshooting User Attribute Retrieval and Transformation

For troubleshooting information, see "Troubleshooting User Attribute Retrieval and Transformation" on page 1023.

Configuring the Access Gateway

The basic Access Gateway configuration procedures consists of the following tasks:

- "Configuring a Reverse Proxy" on page 85
- "Configuring a Public Protected Resource" on page 87
- "Setting Up Policies" on page 88

Configuring a Reverse Proxy

You can protect your Web services by creating a reverse proxy. A reverse proxy acts as the front end to your Web servers in your DMZ or on your intranet. It off-loads frequent requests, thereby freeing up bandwidth and Web server connections. It also increases security because the IP addresses and DNS names of your Web servers are hidden from the Internet. A reverse proxy can be configured to protect one or more proxy services. To configure the Access Gateway, you can create a new configuration as described in this section.

To create a reverse proxy, you must create at least one proxy service with a protected resource. You must supply a name for each of these components. Reverse proxy names and proxy service names must be unique to the Access Gateway because they are configured for global services such as IP addresses and TCP ports. For example, if you have a reverse proxy named `products` and another reverse proxy named `library`, only one of these reverse proxies can have a proxy service named `corporate`.

You can also modify the existing default NAM-RP to match your requirement. The Access Manager Appliance has a default SSL-enabled reverse proxy (NAM-RP). The reverse proxy is associated with a self-signed certificate, which is created during installation of the primary Access Manager

Appliance. To modify the default NAM-RP, click **Devices** > **Access Gateways** > **Edit** > **NAM-RP** in the Administration Console. The default proxy service is NAM-Service. You cannot delete this proxy service and base service. You can modify, enable, disable, rename, and delete the Path-Based Multi-Homing (PBMH), which is created under this proxy service. You can create a new PBMH or Domain-Based Multi-Homing (DBMH) under NAM-service. You can also create a new protected resource, which you can assign it to the newly created PBMH or DBMH. The protected resource, which are not greyed out can also be used to add, delete, modify, enable, and disable paths.

Protected resource names need to be unique to the proxy service, but they don't need to be unique to the Access Gateway because they are always accessed through their proxy service. For example, if you have a proxy service named `account` and a proxy service named `sales`, they both can have a protected resource named `public`.

What You Need To Know	Example	Your Value
DNS name of the Access Gateway	mytest.com	_____
Web server information		
IP address	10.15.70.21	_____
DNS name	mywebserver.com	_____
Names you need to create		
Reverse proxy name	mycompany	_____
Proxy service name	company	_____
Protected resource name	public	_____

This first reverse proxy is used for authentication. You need to configure the proxy service to use the DNS name of the Access Gateway as its **Published DNS Name**, and the Web server and the resource on that Web server need to point to the page you want displayed to the users when they first access your Web site. You can use Access Gateway configuration options to allow this first page to be a public site with no authentication required until the users access the links on the page, or you can require authentication on this first page. Complete the following configuration steps to first configure a protected resource as a public resource and then to modify the configuration to require authentication.

1. In the Administration Console, click **Devices** > **Access Gateways**, **Edit** > **Reverse Proxy / Authentication**.
2. In **Reverse Proxy List**, click **New**, specify a display name for the reverse proxy, then click **OK**.
3. Enable a listening address.

   Listening Address(es): A list of available IP addresses. If the server has only one IP address, only one is displayed and it is automatically selected. If the server has multiple addresses, you can select one or more IP addresses to enable. You must enable at least one address.

   TCP Listen Options: Options for configuring how requests are handled. You cannot set up listening options until you create a proxy service.
4. Ignore the SSL configuration options.

   This basic configuration does not set up SSL. For SSL information, see Chapter 15, "Enabling SSL Communication," on page 819.
5. Configure a listening port.

   Non-Secure Port: Select 80 that is the default port for HTTP.

Secure Port: This is the HTTPS listening port. This port is unused and cannot be configured until you enable SSL.

6 In **Proxy Service List**, click **New**.

7 Specify the following details:

Field	Description
Proxy Service Name	A display name for the proxy service.
Published DNS Name	The DNS name you want the public to use to access your site. For this first proxy server, the DNS name must resolve to the Access Gateway IP address that you selected as the listening address. For the example in Figure 3-2 on page 51, this name would be www.mytest.com.
Web Server IP Address	The IP address of your Web server. This is the Web server with content that you want to share with authorized users and protect from others. In Figure 3-2 on page 51, this is Server 4, whose IP address is 10.15.70.21.
Host Header	The name you want to send in the HTTP header to the Web server. This can either be the published DNS Name (the **Forward Received Host Name** option) or the DNS name of the Web Server (the **Web Server Host Name** option).
Web Server Host Name	The DNS name that the Access Gateway must forward to the Web server. This option is not available if you select **Forward Received Host Name** for the **Host Header** option. The name you use depends upon how you have set up the Web server. If your Web server has been configured to verify that the host name in the header matches its name, you need to specify that name here. In Figure 3-2 on page 51, the Web Server Host Name is mywebserver.com.

8 Click **OK**.

Configuring a Public Protected Resource

The first protected resource in discussed in this configuration is configured to be a public resource.

1 In **Proxy Service List**, click **[Name of Proxy Service]** > **Protected Resources**.

2 In **Protected Resource List**, click **New**, specify a name for the resource, and click **OK**.

3 In the **Contract** field, select **None**.

The **Contract** field must be set to **None**. This is makes this resource a public resource.

4 Configure **URL Path List**.

The default path is `/*`, which allows access to everything on the Web server. Modify this if you need to restrict access to a specific directory on your Web server.

- To delete the default path, select the check box next to the path, then click **Delete**.
- To edit a path in the list, click the path, modify it, then click **OK**.
- To add a path, click **New**, specify the path, then click **OK**. For example, to allow access to the pages in the public directory on the Web server, specify the following path:

```
/public/*
```

5 Click **OK**.

6 In the **Protected Resource List**, verify that the protected resource you created is enabled, then click OK.

7 Click **Devices** > **Access Gateways**.

8 Click **Update** > **OK**.

The system sends configuration changes to the server and writes the configuration to the configuration data store. When the update has completed successfully, the server returns the status of **Current**.

To save the changes to the configuration store without applying them, do not click **Update**. Instead, click **Edit**. If you have pending configuration settings, the **OK** button is active, and the configuration page indicates which services will be updated. Click **OK** to save these changes to the configuration store. The changes are not applied until you **Update** on the Access Gateways page.

9 To update the Identity Server to establish the trust relationship with the Access Gateway, click **Devices** > **Identity Servers** > **Update** > **OK**.

Wait until the **Command** status is **Complete** and the **Health** status is green.

10 (Optional). To test this configuration from a client browser, specify the published DNS name as the URL in the browser. In the example illustrated in Figure 3-2 on page 51, specify the following URL:

```
http://www.mytest.com
```

This must resolve to the published DNS name you specified in Step 7 on page 87, and the user must be connected to the Web server through the Access Gateway.

IMPORTANT: You must not modify the default NAM-Service proxy service.

Setting Up Policies

The Access Gateway lets you retrieve information from your LDAP directory and inject the information into HTML headers, query strings, or basic authentication headers. The Access Gateway can then send this information to the back-end Web servers. Access Manager calls this technology Identity Injection.

This is one of the features within Access Manager that enables single sign-on. Users are prompted for the login credentials for one time, and Access Manager then supplies them for the resources you have configured for Identity Injection.

This section explains how to set up an Identity Injection policy for basic authentication. This policy is assigned to the third directory on your Web server, which is the `basic` directory that your Web server has been configured to require basic authentication before allowing access.

1 In the Administration Console, click **Devices** > **Access Gateways** > **Edit** > **[Reverse Proxy Name]** > **[Proxy Service Name]** > **Protected Resources** > **New**.

2 Configure the resource for the `basic` directory as described in “Prerequisites for Setup” on page 51:

2a For the contract, select **Name/Password - Basic** or **Name/Password - Form**.

2b For the URL path, specify the path to the basic directory (`/basic/*`).

2c Click **OK**.

3 Click **[Protected Resource Name]** > **Identity Injection**.

On a new installation, the list is empty because no policies have been created.

4 In the **Identity Injection Policy List** section, click **Manage Policies**.

5 In the **Policy List** section, click **New**, then specify values for the following fields:

Name: Specify a name for the Identity Injection policy.

Type: Select **Access Gateway: Identity Injection**.

6 Click **OK**.

7 (Optional) Specify a description for the policy.

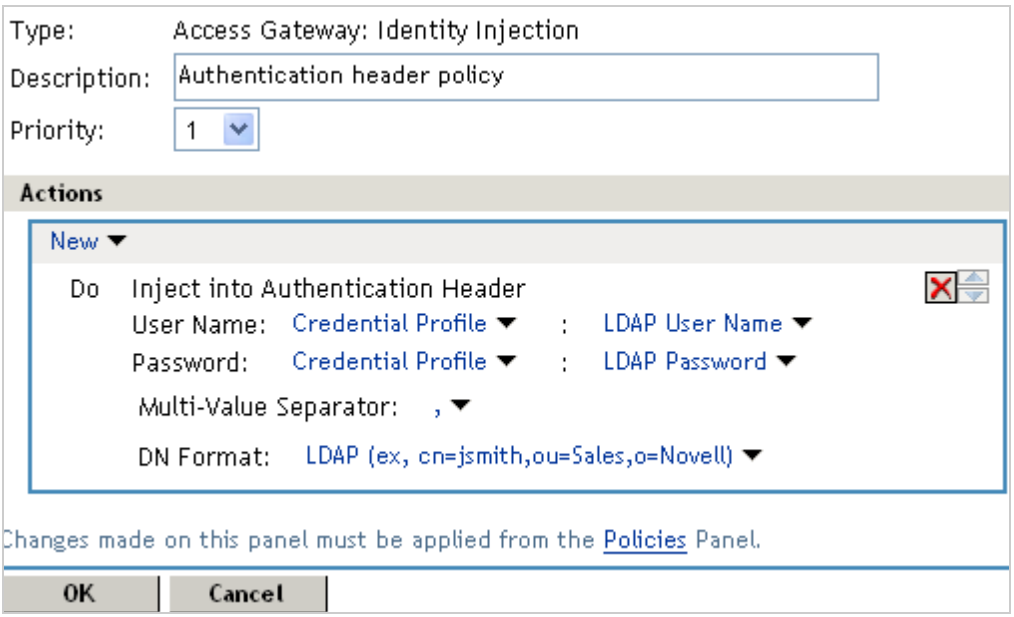
8 In the **Actions** section, click **New** > **Inject into Authentication Header**.

9 Set up the policy for **User Name** and **Password**:

- For **User Name**, select **Credential Profile** and **LDAP Credentials: LDAP User Name**.

  This injects the value of the cn attribute into the header.
- For **Password**, select **Credential Profile** and **LDAP Credentials: LDAP Password**.

The policy must look similar to the following:



10 Click **OK** > **OK** > **Apply Changes** > **Close**.

11 Select the new Identity Injection policy, then click **Enable** > **OK**.

12 Click **Devices** > **Access Gateways** > **Update** > **OK**.

13 To test this configuration from a client browser, specify the published DNS name as the URL in the browser. Click the link to the page that uses basic authentication.

You are prompted to log in. If you have set up Web applications on your Web server that require login, any additional login prompts are hidden from the user and are handled by the identity injection system.

For an example of how Identity Injection policies can be used for single sign-on to the Identity Manager User Application, see "Configuring Access Manager for UserApp and SAML" (http://www.novell.com/coolsolutions/appnote/19981.html).

Access Gateways Clusters

Most of the configuration tasks are same for a single Access Gateway and a cluster of Access Gateways.

Managing the Access Gateway Cluster Configuration

This section describes the tasks that are specific to managing the servers in a cluster:

- “Managing Cluster Details” on page 90
- “Editing Cluster Details” on page 90
- “Applying Changes to the Access Gateway Cluster Members” on page 91

Managing Cluster Details

Use the Cluster Details page to perform general maintenance actions on the selected cluster and to display server information about the selected cluster.

1. In the Administration Console, click **Devices** > **Access Gateways** > **[Cluster Name]**.
2. View the following fields:

   Name: Specifies the name of the cluster.

   Description: Specifies the purpose of the cluster. This is optional, but useful if your network has multiple Access Gateway clusters. If the field is empty, click **Edit** to add a description.

   Primary Server: Indicates which server in the cluster has been assigned to be the primary server.
3. To modify the information, click **Edit**. For more information, see “Editing Cluster Details” on page 90.
4. To select a different Access Gateway to be the primary cluster member, click **Edit**.
5. To modify details about a cluster member, click the server name in the **Cluster member** list.
6. Click **Close**.

Editing Cluster Details

Use the Cluster Detail Edit to change the name of the cluster and assign a different server to be the primary cluster member.

1. In the Administration Console, click **Devices** > **Access Gateways** > **[Cluster Name]** > **Edit**.
2. Modify the following fields:

   Name: Specify a name for the cluster.

   Description: Specify the purpose of the cluster. This is optional, but useful if your network has multiple Access Gateway clusters.

   Primary Server: Indicates which server in the cluster has been assigned to be the primary server. To change this assignment, select the server from the drop-down list.
3. Click **OK**.

Applying Changes to the Access Gateway Cluster Members

When you are configuring services of the Access Gateway, the **OK** button saves the change to browser cache except on the Configuration page. The Configuration page (**Devices** > **Access Gateways** > **Edit**) provides a summary of the changes you have made. The **Cancel Change** column allows you to cancel changes to individual services. When you click **OK**, the changes are saved to the configuration datastore, and you no longer have the option to cancel changes to individual services.

If you don't save the changes to the configuration datastore and your session times out or you log out, any configuration changes that are saved to browser cache are flushed. These changes cannot be applied to other members of the cluster because they are no longer available. To prevent this from happening, save the changes to the configuration datastore.

It is especially important to save the changes to the configuration datastore when you select to update individual members one at a time rather than update all members of the cluster at the same time. Updating members one at a time has the following benefits:

- When you update all servers at the same time, the site goes down until one server has finished updating its configuration. If you update the cluster members one at a time, only the member that is updating its configuration becomes unavailable.
- If you update the servers one at time, you can verify that the changes are behaving as expected. After testing the configuration on one server, you can then apply the saved changes to the other servers in the cluster. If you decide that the configuration changes are not behaving as expected, you can revert to the previously applied configuration. See "Reverting to a Previous Configuration" on page 91.

Some configuration changes cannot be applied to individual cluster members. For a list of these changes, see "Modifications Requiring an Update All" on page 91.

Reverting to a Previous Configuration

If you have updated only one server in the cluster, you can use the following procedure to revert back to the previous configuration.

1. Remove the server that you have applied the configuration changes from the cluster.
2. Access the Configuration page for the cluster, then click **Revert**.

   The servers in the cluster revert to the last applied configuration.
3. Add the removed server to the cluster.

   The server is configured to use the same configuration as the other cluster members.

Modifications Requiring an Update All

When you make the following configuration changes, the **Update All** option is the only option available and your site is unavailable while the update occurs:

- If you change the Identity Server configuration that is used for authentication (**Access Gateways** > **Edit** > **Reverse Proxy/Authentication,** then select a different value for the **Identity Server Cluster** option).
- If you select a different reverse proxy to use for authentication (**Access Gateways** > **Edit** > **Reverse Proxy/Authentication**, then select a different value for the **Reverse Proxy** option).

- If you modify the protocol or port of the authenticating reverse proxy (**Access Gateways** > **Edit** > **Reverse Proxy/Authentication** > **[Name of Reverse Proxy]**, then change the SSL options or the port options).
- If you modify the published DNS name of the authentication proxy service (**Access Gateways** > **Edit** > **Reverse Proxy/Authentication** > **[Name of Reverse Proxy]** > **[Name of First Proxy Service]**, then modify the **Published DNS Name** option).

Protecting Web Resources Through the Access Gateway

The Access Gateway is a reverse proxy server (protected site server) that restricts access to Web-based content, portals, and Web applications that employ authentication and access control policies. It also provides single sign-on to multiple Web servers and Web applications by securely providing the credential information of authenticated users to the protected servers and applications. The Access Gateway lets you simplify, secure, and accelerate your Internet business initiatives.

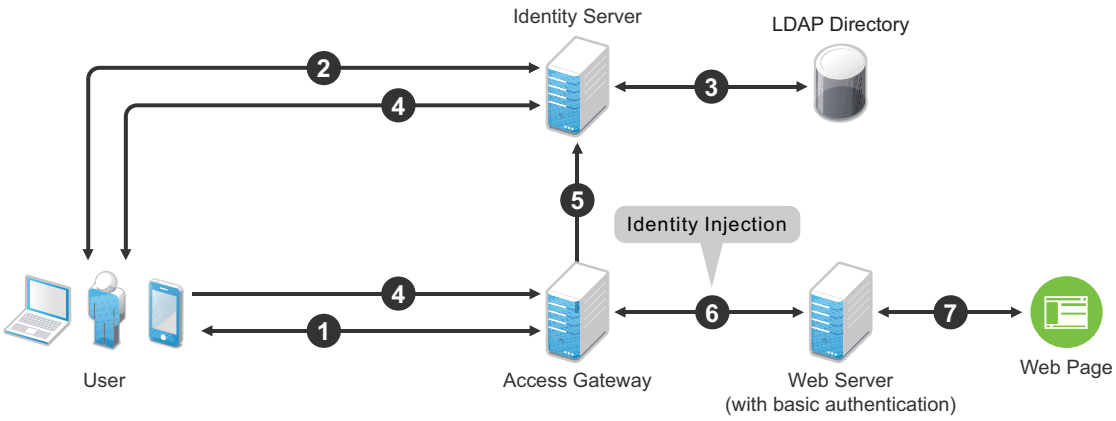
This section describes the following tasks:

- "Configuration Options" on page 92
- "Managing Reverse Proxies and Authentication" on page 94
- "Configuring Web Servers of a Proxy Service" on page 100
- "Configuring Protected Resources" on page 101
- "Configuring HTML Rewriting" on page 113
- "Configuring Connection and Session Limits" on page 130
- "Protecting Multiple Resources" on page 134

Configuration Options

A typical Access Manager Appliance configuration includes an Identity Server with LDAP directories and an Access Gateway with a protected Web server. Figure 3-3 illustrates the process flow that allows an authorized user to access the protected resource on the Web server.

Figure 3-3 *Accessing a Web Resource*

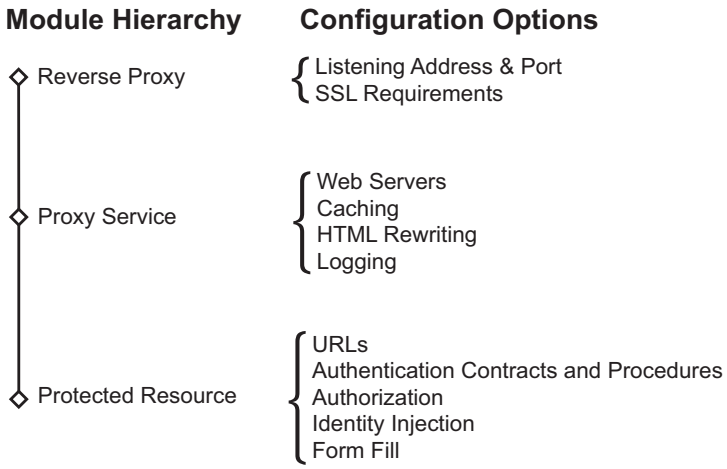


1. The user requests access to a resource protected by the Access Gateway.

2. The Access Gateway redirects the user to the Identity Server, which prompts the user for a username and password.
3. The Identity Server verifies the username and password against an LDAP directory (eDirectory, Active Directory, or Sun ONE).
4. The Identity Server returns an authentication success to the browser and the browser forwards the resource request to the Access Gateway.
5. The Access Gateway verifies that the user is authenticated and retrieves the user's credentials from the Identity Server.
6. The Access Gateway uses an Identity Injection policy to insert the basic authentication credentials in the HTTP header of the request and sends it to the Web server.
7. The Web server grants access and sends the requested page to the user.

When you are setting up the Access Gateway to protect Web resources, you create and configure reverse proxies, proxy services, and protected resources. The following figure illustrates the hierarchy of these modules and the major configuration tasks you perform on each module.

Figure 3-4 *Access Gateway Modules and Their Configuration Options*



This hierarchy allows you to have precise control over what is required to access a particular resource, and also allows you to provide a single sign-on solution for all the resources protected by the Access Gateway. The authentication contract, authentication procedure, Authorization policy, Identity Injection policy, and Form Fill policy are configured at the resource level so that you can enable exactly what the resource requires. This allows you to decide where access decisions are made:

- You can configure the Access Gateway to control access to the resource.
- You can configure the Web server for access control and configure the Access Gateway to supply the required information.
- You can use the first method for some resources and the second method for other resources or use both methods on the same resource.

Managing Reverse Proxies and Authentication

A reverse proxy acts as the front end to your Web servers on your Internet or intranet and off-loads frequent requests, thereby freeing up bandwidth. The proxy also increases security because the IP addresses of your Web servers are hidden from the Internet.

To create a reverse proxy, you must create at least one proxy service with a protected resource. You must supply a name for each of these components. Reverse proxy names and proxy service names must be unique to the Access Gateway because they are configured for global services such as IP addresses and TCP ports. For example, if you have a reverse proxy named `products` and another reverse proxy named `library`, only one of these reverse proxies can have a proxy service named `corporate`.

Protected resource names need to be unique to the proxy service, but they don't need to be unique to the Access Gateway because they are always accessed through their proxy service. For example, if you have a proxy service named `account` and a proxy service named `sales`, they both can have a protected resource named `public`.

The first reverse proxy and proxy service you create are automatically assigned to be the authenticating proxy.

1. In the Administration Console, click **Devices** > **Access Gateways** > **Edit.**

   The **Edit** link is either for a single Access Gateway or for a cluster of Access Gateways.
2. Click **Reverse Proxy / Authentication**.
3. (Conditional) If you have already created at least one reverse proxy, you can view the Embedded Service Provider options and configure some of them:

   Reverse Proxy: Specifies which proxy service is used for authentication. If you have configured only one proxy service, only one appears in the list and it is selected. If you change the reverse proxy that is used for authentication, certificates must be updated to match this new configuration.

   Metadata URL: Displays the location of the metadata.

   Health-Check URL: Displays the location of the health check.

   Logout URL: Displays the URL that you need to use for logging users out of protected resources. This value is empty until you have created at least one reverse proxy and it has been assigned to be used for authentication. If you create two or more reverse proxies, you can select which one is used for authentication, and the logout URL changes to match the assigned reverse proxy.

   If any of your protected resources have a logout page or button, you need to redirect the user's logout request to the page specified by this URL. The Access Gateway can then clear the user's session and log the user out of any other resources that have been enabled for single sign-on. If you do not redirect the user's logout request, the user is logged out of one resource, but the user's session remains active until inactivity closes the session. If the user accesses the resource again before the session is closed, single sign-on reauthenticates the user to the resource, and it appears that the logout did nothing.

   ESP Global Options: Allows you to configure global options for ESP. For more information, see "Configuring ESP Global Options" on page 98.

   Auto-Import Identity Server Configuration Trusted Root: Allows you to import the public key from the Identity Server cluster into the trust store of the Embedded Service Provider. This sets up a trusted SSL relationship between the Embedded Service Provider and the Identity Server. This option is not available until you have selected an **Identity Server Cluster** and have

configured the use of SSL on the Embedded Service Provider of the reverse proxy that is performing authentication (see the **Enable SSL with Embedded Service Provider** option on the Reverse Proxy page).

If the Identity Server cluster is using a certificate created by the Access Manager certificate authority (CA), the public key is automatically added to this trust store, so you do not need to use this option. If the Identity Server cluster is using a certificate created by an external CA, you need to use this option to import the public key into the trust store.

4 (Optional) Configure the proxy settings:

Behind Third Party SSL Terminator: Enable this option if you have installed an SSL terminator between the users and the Access Gateway. This allows the terminator to handle the SSL traffic between the browsers and the terminator. The terminator and the Access Gateway can use HTTP for their communication.

Enable Via Header: Enables the sending of the Via header to the Web server. The Via header contains the DNS name of the Access Gateway and a device ID. It has the following format:

```
Via: 1.1 www.mymag.com (Access Gateway-ag-BFBA9849520DB63B-5)
```

Deselect this option when your Web server does not need this information or does not know what to do with it.

5 (Optional) Configure the cookie settings:

For more information and other options for securing Access Manager cookies, see "Enabling Secure Cookies" on page 790.

Enable Secure Cookies: Enabling this option sets secure keyword on HTTPS request. If you have enabled the **Behind Third Party SSL Terminator** option and also enabled the **Enable Secure Cookies** option, the secure keyword on HTTP and HTTPS requests are set.

WARNING: Do not enable the **Enable Secure Cookies** option if you have both HTTP and HTTPS reverse proxies. The HTTP services become unavailable because authentication requests to the non-HTTP services fail.

Force HTTP-Only Cookie: Forces the Access Gateway to set the HttpOnly keyword, which prevent scripts from accessing the cookie. This helps protect browsers from cross-site scripting vulnerabilities that allow malicious sites to grab cookies from a vulnerable site. The goal of such attacks might be to perform session fixation or to impersonate the valid user.

IMPORTANT: The HttpOnly keyword can prevent applets from loading and can interfere with JavaScript. Do not enable this option if you have the Access Gateway protecting applications that download applets or use JavaScript.

---

6 To create a proxy service, continue with "Creating a Proxy Service" on page 95.

Creating a Proxy Service

1 In the Administration Console, click **Devices** > **Access Gateways** > **Edit** > **Reverse Proxy / Authentication**.

2 In the **Reverse Proxy List**, click **New**, specify a display name for the reverse proxy, then click **OK**.

3 Enable a listening address. Fill in the following fields:

Cluster Member: (Available only if the Access Gateway is a member of a cluster.) Select the server you want to configure from the list of servers. The **Listening Address(es)** and **TCP Listen Options** modifications apply to the selected server. Modifications made to any other options on the page apply to all servers in the cluster.

Listening Address(es): Displays a list of available IP addresses. If the server has only one IP address, only one is displayed and it is automatically selected. If the server has multiple addresses, you can select one or more IP addresses to enable. You must enable at least one address by selecting its check box.

If the Access Gateway is in a cluster, you must select a listening address for each cluster member.

TCP Listen Options: Provides options for configuring how requests are handled between the reverse proxy and the client browsers. You cannot set up the listening options until you create and configure a proxy service. For information about these options, see "Configuring TCP Listen Options for Clients" on page 131.

4 Configure the listening ports:

Non-Secure Port: Specifies the port on which to listen for HTTP requests; the default port for HTTP is 80. Depending upon your configuration, this port might also handle other tasks. These tasks are listed to the right of the text box.

Secure Port: Specifies the port on which to listen for HTTPS requests; the default port for HTTPS is 443.

For information about the SSL options, see "Configuring the Access Gateway for SSL" on page 819.

5 In the **Proxy Service List** section, click **New**.

The first proxy service of a reverse proxy is considered the master (or parent) proxy. Subsequent proxy services can use domain-based, path-based, or virtual multi-homing, relative to the published DNS name of the master proxy service. If you are creating a second proxy service for a reverse proxy, see "Using Multi-Homing to Access Multiple Resources" on page 134.

6 Fill in the fields:

Proxy Service Name: Specify a display name for the proxy service, which the Administration Console uses for its interfaces.

Published DNS Name: Specify the DNS name you want the public to use to access your site. This DNS name must resolve to the IP address you set up as the listening address.

Web Server IP Address: Specify the IP address of the Web server you want this proxy service to manage. You can specify additional Web server IP addresses by clicking the **Web Server Addresses** link when you have finished creating the proxy service.

Host Header: Specify whether the HTTP header must contain the name of the back-end Web server (**Web Server Host Name** option) or whether the HTTP header must contain the published DNS name (the **Forward Received Host Name** option).

Web Server Host Name: Specify the DNS name of the Web server that the Access Gateway must forward to the Web server. If you have set up a DNS name for the Web server and it requires its DNS name in the HTTP header, specify that name in this field. If the Web server has absolute links referencing its DNS name, include this name in this field. If you selected **Forward Received Host Name**, this option is not available.

NOTE: For iChain administrators, the **Web Server Host Name** is the alternate hostname when configuring a Web Server Accelerator.

---

7 Click **OK**.

8 Continue with "Configuring a Proxy Service" on page 97 or select one of the following tasks:

- For information about how to create multiple reverse proxies, see "Managing Multiple Reverse Proxies" on page 143.
- For information about how to create multiple proxy services for a reverse proxy, see "Using Multi-Homing to Access Multiple Resources" on page 134.

Configuring a Proxy Service

A reverse proxy can have multiple proxy services, and each proxy service can protect multiple resources. You can modify the following features of the proxy service:

- Web servers
- HTML rewriting
- Logging
- Protected resources
- Caching

1 To configure a proxy service, click **Access Gateways** > **Edit** > **[Name of Reverse Proxy]** > **[Name of Proxy Service]**.

2 Fill in the following fields:

Published DNS Name: Displays the value that users are currently using to access this proxy service. This DNS name must resolve to the IP address you set up as a listening address on the Access Gateway. You must modify this field only if you have modified the DNS name you want users to use to access this resource.

This name determines the possible values of the **Cookie Domain**.

Description: (Optional). Provides a field where you can describe the purpose of this proxy service or specify any other pertinent information.

Cookie Domain: Specifies the domain for which the cookie is valid.

If one proxy service has a DNS name of www.support.novell.com and the second proxy service has a DNS name of www.developernet.novell.com, the cookie domains are support.novell.com for the first proxy service and developernet.novell.com for the second proxy service. You can configure them to share the same cookie domain by selecting novell.com for each proxy service. Single sign-on between the proxy services is simplified when the proxy services share the same cookie domain.

HTTP Options: Allows you to set up custom caching options for this proxy service. See "Controlling Browser Caching" on page 254.

Advanced Options: Access Gateway Service) Specifies how the proxy service handles specific conditions, such as Web server error pages. If similar options are configured globally, the proxy service configuration overwrites the global setting. For configuration information on the proxy service options, see "Configuring the Advanced Options for a Domain-Based and Path-Based Multi-Homing Proxy Service" on page 269.

3 Click **OK** to save your changes to browser cache.

4 Click **Devices** > **Access Gateways**.

5 To apply your changes, click **Update** > **OK**.

Until this step, nothing has been permanently saved or applied. The **Update** status pushes the configuration to the server and writes the configuration to the configuration data store. When the update has completed successfully, the server returns the status of **Current**.

To save the changes to the configuration store without applying them, do not click **Update**. Instead, click **Edit**. On the Configuration page, click **OK**. The **OK** button on this pages saves the cached changes to the configuration store. The changes are not applied until you click **Update** on the Access Gateways page.

6 Update the Identity Server to accept the new trusted relationship. Click **Identity Servers** > **Update**.

7 Continue with one of the following.

- If the Web server that contains the resources you want to protect does not use the standard HTML port (port 80), you need to configure the Web server. See “Configuring Web Servers of a Proxy Service” on page 100.
- Until you configure a protected resource, the proxy service blocks access to all services on the Web server. To configure a protected resource, see “Configuring Protected Resources” on page 101.

Modifying the DNS Setting for a Proxy Service

1 Get the SSL certificate for the new DNS name.

For more information, see Chapter 11, “Creating Certificates,” on page 799.

2 In the Administration Console, click **Devices** > **Access Gateways**.

3 Edit AG-Cluster and click on any reverse proxy listed under **Reverse Proxy/Authentication**.

4 Change the **Server Certificate** to the new one for your new DNS name.

Ignore any warning displayed about CN name mismatch because the proxy service is not yet updated.

5 Under the **Proxy Service List** tab, click the proxy which DNS name you want to modify.

6 Change the **Published DNS Name** for the proxy service.

NOTE: Changing the published DNS name of the master proxy changes the Identity Server’s base URL also.

---

7 Click **OK** > **OK**.

8 The Cluster Configuration page is displayed.

9 Click **Network Settings** > **Hosts** > *IP address of your system*.

10 Add the new DNS name in the list of host names.

11 Click **OK**.

12 Go to **Access Gateway**.

13 Click **Update All**.

14 When the Access Gateway **Health** turns green, check the Identity Server **Health** and ensure that it is green as well.

Configuring ESP Global Options

When you configure an ESP global option, it gets applied to all Access Gateway ESPs in an Access Gateway cluster.

By default, these options are disabled. To enable these options, you need to remove the pound (#) symbol before it and set a value. After you configure an option, you cannot delete it. However, you can disable it again by adding the pound (#) symbol before it. If you have set a value for an option and want to disable the option, you need to add # before the configured option. After saving the changes, the value for the option is set to the default value. For example, if you have set the value for CLUSTER_COOKIE_DOMAIN as `CLUSTER_COOKIE_DOMAIN .example.com`, add # before CLUSTER_COOKIE_DOMAIN .example.com. After the changes are applied, the option is set to the default value as #CLUSTER_COOKIE_DOMAIN.

NOTE: Access Manager 4.2 onwards, configuring the following options through files is deprecated. You must configure these option by using the Administration Console.

Perform the following steps to configure ESP global options:

1. In the Administration Console, click **Devices** > **Access Gateways** > **Edit** > **Reverse Proxy / Authentication** > **ESP Global Options**.
2. To activate an ESP global option, remove the # symbol before it, configure the value, save it, and then update the Access Gateway. By default, Access Manager displays seven options. You can configure any other options also, if required.

The following table lists the default ESP global options:

ESP Global Option	Description
forceESPSLOHTTP	Set true to enable the front channel logout for the Access Gateway initiated logout. The default value is false. For more information about how to enable front channel logout for the Access Gateway, see "Defining Options for Liberty Identity Provider" on page 461.
httponlyClusterCookie	Set false to disable the HTTPOnly flags for ESP cluster cookies. The default value is true.
CLUSTER_COOKIE_DOMAIN	Set this property to change the Domain attribute for the ESP custer cookie in this format: CLUSTER_COOKIE_DOMAIN .example.com
CLUSTER_COOKIE_PATH	Set this property to change the Path attribute for the ESP custer cookie. The default value is /nesp.
notifysessionTimetoIDP	Set false to disable sending session timeout message to the remote identity provider. The default value is true. For example, see "Configuring the Liberty or SAML 2.0 Session Timeout" on page 437.
RENAME_SESSIONID	Set false to prevent changing the Access Gateway session ID automatically. The default value is true. For example, see "Preventing Automatically Changing Session ID" in the "Securing the Embedded Service Provider Session Cookie on the Access Gateway" on page 791.

ESP Global Option	Description
IS_DISPLAY_AUTH_DONE_PAGE	Set true to enable the Access Gateway to display post-authentication message. The default value is false. For example, see “Enabling the Access Gateway to Display Post-Authentication Message” on page 246.

NOTE: After you configure an ESP option, you cannot revert it to the previous configuration by clicking **Revert** in the Cluster Configuration page (**Access Gateway** > **Edit** > **Revert**).

Configuring Web Servers of a Proxy Service

The Web server configuration determines how the Access Gateway handles connections and packets between itself and the web servers.

IMPORTANT: For caching to work correctly, the Web servers must be configured to maintain a valid time. They must be configured to use an NTP server.

1. Click **Access Gateways** > **Edit** > **[Name of Reverse Proxy]** > **[Name of Proxy Service]** > **Web Servers**.
2. Specify the hostname that is placed in the HTTP header of the packets being sent to the Web servers. In the **Host Header** field, select one of the following:
   - **Forward Received Host Name:** Indicates that you want the HTTP header to contain the published DNS name that the user sent in the request.
   - **Web Server Host Name:** Indicates that you want the published DNS name that the user sent in the request to be replaced by the DNS name of the Web server. Use the **Web Server Host Name** field to specify this name. You can also append the port number to the **Web Server Host Name** field. For example, <*web server hostname*>:<*web server port number*>.
3. Select **Error on DNS Mismatch** to have the proxy determine whether the proxy service must compare the hostname in the DNS header that came from the browser with the DNS name specified in the **Web Server Host Name** option. The value in the parentheses is the value that comes in the header from the browser.

   If you enable this option and the names don't match, the request is not forwarded to the Web server. Instead, the proxy service returns an error to the requesting browser. This option is only available when you select to send the **Web Server Host Name** in the HTTP header.

   NOTE: The **Error on DNS Mismatch** option does not work in the following scenarios:
   - If the option is enabled in a protected resource.
   - If the option is enabled in a master host based service, and disabled in a path-based child services, then the Access Gateway does a strict check of DNS match for path-based child.
4. If your browsers are capable of sending HTTP 1.1 requests, configure the following fields to match your Web servers:

   Enable Force HTTP 1.0 to Origin: Indicates whether HTTP 1.1 requests from browsers are translated to HTTP 1.0 requests before sending them to the Web server. If your browsers are sending HTTP 1.1 requests and your Web server can only handle HTTP 1.0 requests, you must enable this option.

When the option is enabled, the Access Gateway translates an HTTP 1.1 request to an HTTP 1.0 request.

Enable Session Stickiness: Selecting this option makes the proxy server to use the same Web server for all fills during a session.

5 To enable SSL connections between the proxy service and its Web servers, select **Connect Using SSL**. For configuration information for this option, **Web Server Trusted Root**, and **SSL Mutual Certificate**, see "Configuring SSL between the Proxy Service and the Web Servers" on page 828.

6 In the **Connect Port** field, specify the port that the Access Gateway must use to communicate with the Web servers. The following table lists some default port values for common types of Web servers.

Server Type	Non-Secure Port	Secure Port
Web server with HTML content	80	443
WebSphere	9080	9443
JBoss	8080	8443

7 To control how idle and unresponsive Web server connections are handled and to optimize these processes for your network, select **TCP Connect Options**. For more information, see "Configuring TCP Connect Options for Web Servers" on page 132.

8 To add a Web server, click **New** in the **Web Server List** and specify the IP address or the fully qualified DNS name of the Web server.

The Web servers added to this list must contain identical Web content. Configuring your system with multiple servers with the same content adds fault tolerance and increases the speed for processing requests. For more information about this process, see "Configuring Web Servers" on page 133.

9 To delete a Web server, select the Web server, then click **Delete**.

This deletes the Web server from the list so that the Access Gateway no longer sends requests to the deleted Web server. At least one Web server must remain in the list. You must delete the proxy service to remove the last server in the list.

NOTE: Do not remove all configured Web servers to the cluster if any of the cluster member does not have at least one Web server configured.

10 To save your changes to browser cache, click **OK**.

11 To apply your changes, click the **Access Gateways** link, then click **Update** > **OK**.

Configuring Protected Resources

A protected resource configuration specifies the directory (or directories) on the Web server that you want to protect. The protected resource configuration specifies the authorization procedures and the policies that must be used to enforce protection. The authentication procedures and the policies (Authorization, Identity Injection, and Form Fill) enable the single sign-on environment for the user. The type of protection a resource requires depends upon the resource, the Web server, and the conditions you define for the resource.

You can select from the following types of protection:

Authentication Procedures: Specifies the type of credentials the user must use to log in (such as name and password or secure name and password). You can select **None** for the procedure, which allows the resource to be a public resource, with no login required.

In addition to selecting the contract, you can also configure how the authentication procedure handles subsequent authentication requests from an application.

Authorization Policy: Specifies the conditions a user must meet to be allowed access to a protected resource. You define the conditions, and the Access Gateway enforces the Authorization policies. For example, you can assign roles to your users, and use these roles to grant and deny access to resources.

Identity Injection Policy: Specifies the information that must be injected into the HTTP header. If the Web application has been configured to look for certain fields in the header and the information cannot be found, the Web application determines whether the user is denied access or redirected. The Web application defines the requirements for Identity Injection. The Identity Injection policies allow you to inject the required information into the header.

Form Fill Policy: Allows you to manage forms that Web servers return in response to client requests. Form fill allows you to prepopulate fields in a form on first login and then securely save the information in the completed form to a secret store for subsequent logins. The user is prompted to reenter the information only when something changes, such as a password.

These policies allow you to design a custom access policy for each protected resource:

- Resources that share the same protection requirements can be configured as a group. You set up the policies, and then add the URLs of each resource that requires these policies.
- A resource that has specialized protection requirements can be set up as a single protected resource. For example, a page that uses Form Fill is usually set up as a single protected resource.

After configuring a protected resource, you can bookmark it. You cannot bookmark a login page that is used in a federation setup.

To configure a protected resource:

1. In the Administration Console, click **Devices** > **Access Gateways > Edit > [Name of Reverse Proxy] > [Name of Domain-Based Proxy Service or Primary Proxy Service] > Protected Resources**.

   The **Resource View** of the **Protected Resource List** is used to create new protected resources or manage existing protected resources. The **Policy View** is used to see which policies are being used by multiple protected resources. For more information about the **Policy View**, see “Assigning a Policy to Multiple Protected Resources” on page 112.

2. Select one of the following actions:

   New: To create a new protected resource, click this option and specify a display name for the resource. For configuration information, see “Setting Up a Protected Resource” on page 103.

   Delete: To delete a protected resource, select a protected resource, then click **Delete**.

   Enable: To enable a resource so that the Access Gateway protects it, select a protected resource, then click **Enable**.

   Disable: To disable protection for a resource, select a protected resource, then click **Disable**. After a resource is disabled, its path no longer has special protection. For example, you can set up a resource that allows access to all pages (for example /*) and another resource with special protection for a subpath. If you disable the subpath, make sure the security requirements of the /* resource are sufficient for the subpath.

Also, when a protected resource is disabled, the resource no longer shows up in the Path List for a path-based multi-homing proxy.

3 Select the name of a protected resource to perform the following tasks:

- “Configuring an Authentication Procedure for Non-Redirected Login” on page 105
- “Assigning an Authorization Policy to a Protected Resource” on page 107
- “Assigning an Identity Injection Policy to a Protected Resource” on page 107
- “Assigning a Form Fill Policy to a Protected Resource” on page 108
- “Assigning a Timeout Per Protected Resource” on page 110

Setting Up a Protected Resource

1 Click **Access Gateways** > **Edit** > **[Name of Reverse Proxy]** > **[Name of Proxy Service] > Protected Resources**.

2 Either click the name of an existing resource or click **New**, then specify a display name for the resource.

3 (Optional) Specify a description for the protected resource. You can use it to briefly describe the purpose for protecting this resource.

4 Select the type of contract to use for the authentication procedure. The contract determines the information a user must supply for authentication. By default, the Administration Console allows you to select from the following contracts and options when specifying whether a resource requires an authentication contract:

None: If you want to allow public access to the resource and not require an authentication contract, select **None**.

Any Contract: If the user has authenticated, this option allows any contract defined for the Identity Server to be valid, or if the user has not authenticated, it prompts the user to authenticate, using the default contract assigned to the Identity Server configuration.

Name/Password - Basic: Specifies basic authentication over HTTP, using a standard login pop-up provided by the Web browser.

Name/Password - Form: Specifies a form-based authentication over HTTP or HTTPS, using the Access Manager login form.

Secure Name/Password - Basic: Specifies basic authentication over HTTPS, using a standard login pop-up provided by the Web browser.

Secure Name/Password - Form: Specifies a form-based authentication over HTTPS, using the Access Manager login form.

The contract also determines the session timeout for inactive connections. If you have some resources that need to time out quickly to protect sensitive data and other resources that don’t need this kind of protection, you need to configure contracts for these resources. For more information about this feature, see “Assigning a Timeout Per Protected Resource” on page 110.

If no contracts are available, you have not configured a relationship between the Access Gateway and the Identity Server. See “Managing Reverse Proxies and Authentication” on page 94.

5 (Conditional) To modify how the authentication procedures are handled for a specific resource and contract, click the **Edit Authentication Procedures** icon.

For configuration information, see “Configuring an Authentication Procedure for Non-Redirected Login” on page 105.

6 Configure the **URL Path**.

The default path is /*, which indicates everything on the Web server. Modify this if you need to restrict access to a specific directory on your Web server. If you have multiple directories on your Web server that require the same authentication contract and access control, add each directory as a URL path.

New: To add a path, click **New**, specify the path, then click **OK**. For example, to allow access to all the pages in the public directory on the Web server, specify the following path:

```
/public/*
```

To allow access to all the files in a directory, but not to the subdirectories and their files, specify the following:

```
/?
```

```
/public/?
```

The `/?` allows access to the root directory, but not the subdirectories. The `/public/?` allows access to the files in the public directory, but not the subdirectories.

To allow access to files of a specific type, specify the following:

```
/public/*.pdf
```

This allows access to all the files in the public directory that have a PDF extension. Access to other file types and subdirectories is denied.

To use this protected resource to protect a single page, specify the path and the filename. For example, to protect the `login.html` page in the `/login` directory, specify the following:

```
/login/login.html
```

This is the type of URL path you want to specify when you create a Form Fill policy for a protected resource. The **URL Path List** normally contains only this one entry. If you have multiple pages that the Form Fill policy applies to, list each one separately in the list. For optimum speed, you want the Access Gateway to be able to quickly identify the page and not search other pages to see if the policy applies to them.

For more information about how a user's request is matched to a protected resource, see "Understanding URL Path Matching" on page 104.

For more information about using a query string, see "Using a Query String in the URL Path" on page 105.

Modify: To modify a path, click the path link, then modify the **URL Path**.

Delete: To delete a path, select the path, then click **Delete**.

7 Click **OK**.

8 In the **Protected Resource List**, ensure that the protected resource you created is enabled.

9 (Optional) To add policies for protecting this resource, continue with one of the following:

- "Assigning an Authorization Policy to a Protected Resource" on page 107
- "Assigning an Identity Injection Policy to a Protected Resource" on page 107
- "Assigning a Form Fill Policy to a Protected Resource" on page 108
- "Assigning a Policy to Multiple Protected Resources" on page 112

10 To apply your changes, click the **Access Gateways** link, then click **Update** > **OK**.

Understanding URL Path Matching

The URL path determines which protected resource is used for a user request. Suppose you create one protected resource with the following URL paths:

```
/*
/test/*
/test/
```

You create a second protected resource with the following path:

```
/test/*.php
```

Users then send the following paths in their access requests:

```
/test/
/test/1/2/3/file.php
/file.php
/test/file.php
/test/file.php?param1=1234
```

The first three requests (`/test/`, `/test/1/2/3/file.php`, and `/file.php`) match the first protected resource, and the last two requests (`/test/file.php` and `/test/file.php?param1=1234`) match the second protected resource.

You then add the following URL path to the first protected resource:

```
/test/?
```

This URL path in the first protected resource causes all the requests to match the first protected resource, and the second protected resource is ignored. The ? wildcard, which matches all content in the current directory, takes precedence over the more specific wildcard (*.php).

Using a Query String in the URL Path

You can specify a query string in the URL path of a protected resource. For example:

URL path: `/test/index.html?test=test`

When the requested URL has a query string, the Access Gateway searches for a protected resource with a matching URL path and query string. If it can't find a match, the request returns a `resource not found` error.

The Access Gateway Service does not have this option. If you want the query string ignored, you must remove it from the URL path of the protected resource.

Configuring an Authentication Procedure for Non-Redirected Login

When a contract is created, it is assigned an authentication procedure that allows the user to be redirected to the Identity Server for authentication. Some applications, such as AJAX and WebDAV applications, do not support redirection for authentication. You can change the authentication behavior of a contract so that redirection does not occur.

When non-redirected login is enabled, the Access Gateway prompts the user to supply basic authentication credentials. The SOAP back channel between the Access Gateway and the Identity Server is used to complete the authentication on the user's behalf rather than a redirect. The SOAP back channel is also used for the session renewals.

Non-redirected login has the following restrictions:

- **Password Expiration Services:** When you modify the authentication procedures to use non-redirected login, you cannot also use a password expiration service. Even when the **Password expiration servlet** and **Allow user interaction** options are configured, users are not redirected when their passwords are expiring and they are not prompted to change their passwords.

- **Locked Shared Secrets:** When non-redirected login is enabled, users are not prompted for their passphrase for locked shared secrets.
- **Session Limits:** Non-redirected login can cause the user to create more than one session with the Identity Server because the SOAP back channel uses a different process than authentication requests that are directed to the Identity Server. Therefore, do not limit your users to one session. Session limits are set by clicking **Devices** > **Identity Servers** > **Edit**.

If the contract you are going to use for non-redirected login is also assigned to protected resources that do not require non-redirected login, you must create a new authentication procedure for the resource requiring non-redirected login. Multiple authentication procedures can be configured to use the same contract.

To configure an authentication procedure:

1. Click **Access Gateways** > **Edit** > **[Name of Reverse Proxy]** > **[Name of Proxy Service]** > **Protected Resources** > **[Name of Protected Resource]**.
2. On the Authentication Procedure line, click the **Edit Authentication Procedure** icon.

   The Authentication Procedure List displays all available contracts, the name of the authentication procedure they are assigned to, the protected resources that the authentication procedure has been assigned to, and whether the procedure has been enabled for non-redirected login.
3. Select one of the following actions:
   - To create an new authentication procedure, click **New**, specify a name, then click **OK**. Continue with Step 4.
   - To modify an existing authentication procedure, click the name of the procedure. Continue with Step 4.
   - To delete an existing authentication procedure, select the procedure, then click **Delete**. Continue with Step 7.

     If the procedure is used by a resource, it cannot be deleted until it is not being used to protect resources. An authentication procedure must exist for each contract. If you delete an authentication procedure for a contract without also deleting the contract, the system automatically re-creates an authentication procedure for the contract.
4. To specify the method for obtaining the credentials, fill in the following fields:

   Contract: Select the contract that you want to use for this protected resource. This needs to be a contract that supports basic authentication credentials such as Name/Password- Basic or Secure Name/Password-Basic. You can also configure Non-Redirected Login with a Kerberos contract.

   Non-Redirected Login: Select this option to use the SOAP back channel to verify the user's credentials rather than a redirected request to the Identity Server.

   Realm: Specify a name that your users can use to identify the site that they are authenticating to. This could be your company name or the name of the application. The realm is displayed as a heading when the application requests a basic authentication.

   Redirect to Identity Server When No Authentication Header Is Provided: The response must provide an authentication header. If the first request does not contain the authentication header, you can select this option to allow the first request to be redirected to the Identity Server.
5. Click **OK**.
6. For the Authentication Procedure, select the authentication procedure you created or modified in Step 4.
7. Click **OK**.
8. Click **Devices** > **Access Gateways**, then update the Access Gateway.

9 (Optional) For some configuration scenarios that use this feature, see

- “Configuring Protected Resource for a SharePoint Server” on page 174
- “Configuring a Protected Resource for a SharePoint Server with an ADFS Server” on page 174
- “Configuring a Protected Resource for Outlook Web Access” on page 177
- “Configuring a Protected Resource for a Novell Vibe 3.3 Server” on page 180

Assigning an Authorization Policy to a Protected Resource

An Authorization policy specifies conditions that a user must meet in order to access a resource. The Access Gateway enforces these conditions. The policy can specify the criteria a user must meet either to allow access or to deny access.

1 Click **Access Gateways** > **Edit > [Name of Reverse Proxy] > [Name of Proxy Service]** > **Protected Resources** > **[Name of Protected Resource]** > **Authorization**.

The **Authorization Policy List** contains all the Access Gateway Authorization policies that have been created on this Administration Console for the selected policy container.

2 Select one of the following:

- To enable an existing policy, select the policy, then click **Enable**. Continue with Step 4.
- To disable an existing policy, select the policy, then click **Disable**. Continue with Step 4.
- To edit an existing policy, click the name of the policy. Remember that policies can be assigned to multiple protected resources. If you modify the policy, you are also affecting how this policy protects those resources. For configuration information, see “Authorization Policies” on page 642.

  When you have completed your policy modifications, continue with Step 4.
- To create a new policy, click **Manage Policies**. On the Policies page, click **New**, specify a display name, select **Access Gateway: Authorization** as the type, then click **OK**. For configuration information, see “Creating Access Gateway Authorization Policies” on page 653.

  When you have created your policy, continue with Step 3.

3 To enable the policy you just created, select the policy, then click **Enable**.

Only the policies that are enabled are applied to this resource. All available Authorization policies are listed. If you use the same policy for multiple protected resources, use the policy description field to indicate this.

4 To save your changes to the browser cache, click **OK**.

5 To apply the changes, click the **Access Gateways** link, then click **Update** > **OK**.

Assigning an Identity Injection Policy to a Protected Resource

The Web application defines the requirements for Identity Injection. If a Web application has been configured to look for certain fields in the header and the information cannot be found, the Web application determines whether the user is denied access, granted access, or redirected. You configure an Identity Injection policy to inject into the HTTP header the information that the Web application requires.

1 Click **Access Gateways** > **Edit > [Reverse Proxy Name] > [Name of Proxy Service]** > **Protected Resources** > **[Name of Protected Resource]** > **Identity Injection**.

The **Identity Injection Policy List** contains all the Identity Injection policies that have been created on this Administration Console for the selected policy container.

2 Select one of the following:

- To enable an existing policy, select the policy, then click **Enable**. Only the policies that are enabled are applied to this resource. Continue with Step 4.
- To disable an existing policy, select the policy, then click **Disable**. Continue with Step 4.
- To edit an existing policy, click the name of the policy. Remember that policies can be assigned to multiple protected resources. If you modify the policy, you are also affecting how this policy protects those resources. For configuration information, see Chapter , “Identity Injection Policies,” on page 691.

  When you have finished your policy modifications, continue with Step 4.
- To create a new policy, click **Manage Policies**. On the Policies page, click **New**, specify a display name, select **Access Gateway: Identity Injection** as the type, then click **OK**. For configuration information, see Chapter , “Identity Injection Policies,” on page 691.

  When you have created your policy, continue with Step 3.

3 To enable the policy you just created, select the policy, then click **Enable**.

Only the policies that are enabled are applied to this resource. If you use the same policy for multiple protected resources, use the policy description field to indicate this.

4 To save your changes to the browser cache, click **OK**.

5 To apply your changes, click the **Access Gateways** link, then click **Update** > **OK**.

IMPORTANT: If you enable an Identity Injection policy for a protected resource that has been assigned to use a contract that does not prompt the user for a password and the Identity Injection policy injects the user’s password, single sign-on cannot be enabled because the password is not available. However, you can create a contract that retrieves the user’s password when the user is not prompted for a password when authenticating. See “Password Retrieval” on page 323.

Assigning a Form Fill Policy to a Protected Resource

Some client requests cause the Web server to return a form. Sometimes this form contains a request to log in. If you create a Form Fill policy, you can have the Access Gateway fill in the form. When a user first logs in, the Access Gateway prepopulates some fields and prompts the users for the others. The Access Gateway securely saves the information, so that on subsequent logins, the Access Gateway can fill in the form. The user is only prompted to fill in the form when something changes, such as a password expiring.

Form Fill uses two components: the HTML form and the Form Fill policy. The HTML form is created with HTML tags and consists of form elements such as fields, menus, check boxes, and buttons. The Form Fill policy is created by specifying the following:

- Which information is entered automatically and not displayed to the user.
- Which information is displayed so that the user, at least the first time, can enter the information.
- What is done with the information (for example, whether it is saved so that the user does not need to enter it when accessing the form again).

You must create the policy before you can assign it to a resource (see Chapter , “Form Fill Policies,” on page 711). To assign a Form Fill policy to a protected resource:

1 In the Administration Console, click **Devices** > **Access Gateways** > **Edit** > **[Reverse Proxy Name]** > **[Name of Proxy Service]** > **Protected Resources** > **[Name of Protected Resource]**.

2 Examine the entries in the **URL Path List**.

Ideally, the URL to which you are assigning a Form Fill policy must be a single HTML page or a few HTML pages. If possible, it must not be a URL that ends in a wildcard (for example, an asterisk) and therefore matches many pages.

IMPORTANT: When the URL ends in a wildcard, the Access Gateway must search each page that matches the URL and check to see if it contains the form. This adds extra processing overhead for all the pages that match the URL, but do not contain the form. For more information about the performance problems this can cause, see Chapter , “Form Fill Policies,” on page 711.

---

3 (Conditional) If the URL is not specific, click the name of the path and modify it.

4 Click **Form Fill**.

The **Form Fill Policy List** contains all the Form Fill policies that have been created on this Administration Console for the selected policy container.

5 Select one of the following:

- To enable an existing policy, select the policy, then click **Enable**. Only the policies that are enabled are applied to this resource. Continue with Step 7.
- To disable an existing policy, select the policy, then click **Disable**. Continue with Step 7.
- To edit an existing policy, click the name of the policy. Remember that policies can be assigned to multiple protected resources. If you modify the policy, you are also affecting how this policy protects those resources. For configuration information, see Chapter , “Form Fill Policies,” on page 711.

  When you have finished the policy modifications, continue with Step 7.
- To create a new policy, click **Manage Policies**. On the Policies page, click **New**, specify a display name, select **Access Gateway: Form Fill** as the type, then click **OK**. For configuration information, see Chapter , “Form Fill Policies,” on page 711.

  When you have created your new policy, continue with Step 6.

6 To enable the policy you just created, select the policy, then click **Enable**.

Only the policies that are enabled are applied to this resource. If you use the same policy for multiple protected resources, use the policy description field to indicate this.

7 To save your changes to the browser cache, click **OK**.

8 To apply your changes, click the **Access Gateways** link, then click **Update** > **OK**.

IMPORTANT: If you enable a Form Fill policy for a protected resource that has been assigned to use a contract that does not prompt the user for a password and the Form Fill policy contains a field for the user’s password, single sign-on cannot be enabled because the password is not available. To enable single sign-on, you need to use an Authentication class that retrieves the user’s password and injects it into the user’s credentials when the user authenticates using a non-password method such as X.509, RADIUS, smart card, or Kerberos. For information about such a class, see “Password Retrieval” on page 323.

Assigning a Timeout Per Protected Resource

If all your resources are using the same contract and you want them all to have the same timeout for inactivity, you set the **Authentication Timeout** option on the contract to the required limit and leave the **Activity Realm** option blank. The user logs in, and activity by the user on any resource keeps the user's session active. The user is prompted to reauthenticate only when the user has no activity on any resources for longer than the authentication timeout value.

If you have some resources that require a shorter timeout than other resources, you need to balance the need for single sign-on with the timeout requirements:

- To strictly enforce a timeout, the resource needs to be assigned to a custom contract.
- To preserve single sign-on, resources need to be assigned to the same contract.

The protected resource is assigned to use a contract, and the timeout is assigned to the contract. For information about how to configure the contract, see "Configuring Authentication Contracts" on page 288.

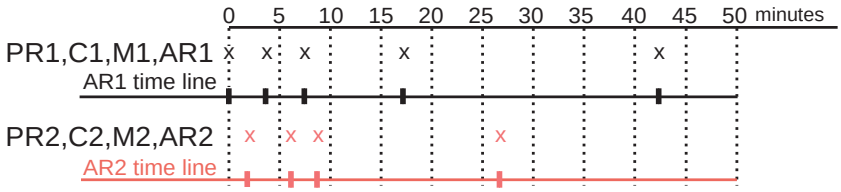
The following sections describe four configuration scenarios and the user experience that they create.

Scenario 1: If strictly adhering to the timeout value is more important than preserving the session or single sign-on, configure your resources as follows:

- Protected resource 1 (PR1) is configured to use contract 1 (C1), which has been created from method 1 (M1) and placed in its own activity realm (AR1). For this scenario you set the authentication timeout to 30 minutes.
- Protected resource 2 (PR2) is configured to use contract 2 (C2), which has been created from method 2 (M2) and placed in its own activity realm (AR2). For this scenario, you set the authentication timeout to 15 minutes.

With this scenario, the user is prompted to log in when accessing PR1 and when accessing PR2. Each resource has its own time line, because each resource belongs to its own activity realm. Figure 3-5 The figure below illustrates this scenario.

Figure 3-5 *Login Requirements with Separate Methods and Separate Activity Realms*



After authenticating to both resources and remaining active on both resources for the first 10 minutes, the sessions remain active. The user then stays active on PR1 without accessing PR2 for over 15 minutes. The AR1 time line is updated with this activity. The AR2 time line is not updated. When the user accesses PR2 after more than 15 minutes of inactivity on the AR2 time line, the user is prompted to authenticate. The user then returns to PR1 after over 20 minutes of inactivity, but AR1 time line shows activity within the 30-minute timeout. The user is granted access and does not need to log in again to access PR1.

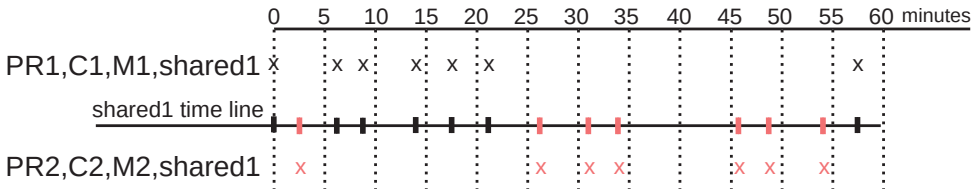
In this scenario, the resources are independent of each other and do not influence each other's timeout limits.

Scenario 2: If you are willing to allow a resource to influence the timeout of another resource, configure your resources as follows:

- Protected resource 1 (PR1) is configured to use contract 1 (C1), which has been created from method 1 (M1) and placed in a shared activity realm (shared1). For this scenario you set the authentication timeout to 30 minutes.
- Protected resource 2 (PR2) is configured to use contract 2 (C2), which has been created from method 2 (M2) and placed in a shared activity realm (shared1). For this scenario, you set the authentication timeout to 15 minutes.

With this scenario, the user is prompted to log in when accessing PR1 and when accessing PR2. Activity at either resource updates the shared1 time line. Figure 3-6 illustrates this scenario.

Figure 3-6 *Login Requirements for Separate Methods with a Shared Activity Realm*



As long as the user is active on PR1, the user's session to PR2 remains active. After 20 minutes of activity on PR1, the user returns to PR2. The user is allowed access and does not need to log in because the shared1 time line shows activity within the last 5 minutes. The user remains active on PR2 for over 30 minutes, then accesses PR1. Again, the shared1 time line shows activity within the last 5 minutes, so the user is granted access to PR1 without logging in again.

With this configuration, activity at other resources influences the time limits so that they are not strictly enforced.

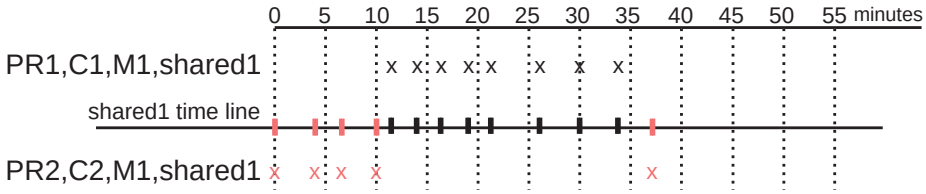
Scenario 3: If single sign-on is more important than strictly enforcing a timeout value, NetIQ recommends that you configure all contracts to have the same authentication timeout value.

If you configure your resources as follows, you might not get the behavior you require:

- Protected resource 1 (PR1) is configured to use contract 1 (C1), which has been created from method 1 (M1) and placed in a shared activity realm (shared1). For this scenario you set the authentication timeout to 30 minutes.
- Protected resource 2 (PR2) is configured to use contract 2 (C2), which has been created from method 1 (M1) and placed in a shared activity realm (shared1). For this scenario, you set the authentication timeout to 15 minutes.

Because C1 and C2 are created from the same method (M1), the user does not need to log in twice to access both resources. Logging in to one resource allows them access to the other resource. Figure 3-7 illustrates this scenario.

Figure 3-7 *Login Requirements for Shared Methods and Shared Realms*



The user first logs in to PR2 and is active for 10 minutes. The shared1 time line gets updated with this activity. When the user requests access to PR1, the user is granted access without being prompted for credentials. The user is then active on PR1 for over 20 minutes. When the user requests access to PR2, even though the user has been inactive on this resource for over 20 minutes, the user is granted access because the time line shows activity within the last five minutes.

With this configuration, PR2 does not time out as long as the user remains active on PR1. However, when the user goes inactive on both PR2 and PR1 for over 15 minutes and the user requests access to PR1, the time line shows no activity within the time limit specified for PR2 and the user is prompted to log in.

Scenario 4: NetIQ does not recommend that you set different authentication timeouts on contracts and then use the Any contract option for protected resources. If you want to use the Any contract, then you must set the authentication timeout to the same value on all contracts. If the timeouts are not the same, you cannot consistently predict what timeouts are being applied to the various protected resources. For example, the user requests access to a resource that is protected with a contract with a short timeout. The user logs in, then accesses resources that use the Any contract option. All of these resources are assigned a short timeout.The user then goes inactive and the session times out. The user then requests access to a resource with a contract with a long timeout. The user logs in, and after a few minutes, accesses same resources protected with the Any contract option. These resources are now assigned the long timeout value.

Assigning a Policy to Multiple Protected Resources

If you have created multiple protected resources that need to be protected by the same policy or policies, you can use the policy view to assign a policy to multiple protected resources. However, the protected resources must belong to the same proxy service.

1. In the Administration Console, click **Devices** > **Access Gateways** > **Edit** > **[Reverse Proxy Name]** > **[Name of Proxy Service]** > **Protected Resources**.
2. Select the **Policy View**.
3. Select the **Used By** link of the policy you want to assign to multiple resources.

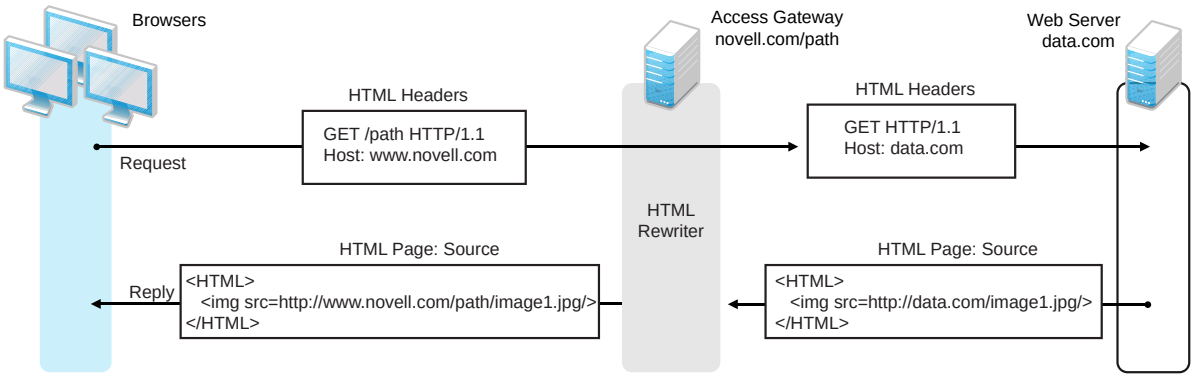
   The **Policy** and **Policy Container** fields identify the policy. The **Protected Resource Policy Usage List** displays the protected resources defined for this proxy service and indicates which resources the policy has been enabled on.
4. To enable the policy for multiple resources, either select them one by one or click **Name** to select all of them, then click **Enable**. To disable a policy for a resource, select the resource, then click **Disable**.
5. To save your changes to browser cache, click **OK**.
6. To apply your changes, click the **Access Gateways** link, then click **Update** > **OK**.

Configuring HTML Rewriting

Access Gateway configurations generally require HTML rewriting because the Web servers are not aware that the Access Gateway machine is obfuscating their DNS names. URLs contained in their pages must be checked to ensure that these references contain the DNS names that the client browser understands. On the other end, the client browsers are not aware that the Access Gateway is obfuscating the DNS names of the resources they are accessing.

The URL requests coming from the client browsers that use published DNS names must be rewritten to the DNS names that the Web servers expect. Figure 3-8 illustrates these processes.

Figure 3-8 *HTML Rewriting*



The following sections describe the HTML rewriting process:

- “Understanding the Rewriting Process” on page 113
- “Specifying DNS Names to Rewrite” on page 115
- “Defining the Requirements for the Rewriter Profile” on page 118
- “Configuring the HTML Rewriter and Profile” on page 125
- “Creating or Modifying a Rewriter Profile” on page 127
- “Disabling the Rewriter” on page 128

Understanding the Rewriting Process

The Access Gateway needs to rewrite URL references under the following conditions:

- To ensure that URL references contain the proper scheme (HTTP or HTTPS).

  If your Web servers and Access Gateway machines are behind a secure firewall, you might not require SSL sessions between them, and only require SSL between the client browser and the Access Gateway. For example, an HTML file being accessed through the Access Gateway for the Web site `novell.com` might have a URL reference to `http://novell.com/path/image1.jpg`. If the reverse proxy for `novell.com/path` is using SSL sessions between the browser and Access Gateway, the URL reference `http://novell.com/path/image1.jpg` must be rewritten to `https://novell.com/path/image1.jpg`. Otherwise, when the user clicks the HTTP link, the browser must change from HTTP to HTTPS and establish a new SSL session.
- To ensure that URL references containing private IP addresses or private DNS names are changed to the published DNS name of the Access Gateway or hosts.

For example, suppose that a company has an internal Web site named `data.com`, and wants to expose this site to Internet users through the Access Gateway by using a published DNS name of `novell.com`. Many of the HTML pages on this Web site have URL references that contain the private DNS name, such as `http://data.com/imagel.jpg`. Because Internet users are unable to resolve `data.com/imagel.jpg`, links using this URL reference would return DNS errors in the browser.

The HTML rewriter can resolve this issue. The **DNS name** field in the Access Gateway configuration is set to `novell.com`, which users can resolve through a public DNS server to the Access Gateway. The rewriter parses the Web page, and any URL references matching the private DNS name or private IP address listed in the Web server address field of the Access Gateway configuration are rewritten to the published DNS name `novell.com` and the port number of the Access Gateway.

Rewriting URL references addresses two issues: 1) URL references that are unreachable because of the use of private DNS names or IP addresses are now made accessible and 2) Rewriting prevents the exposure of private IP addresses and DNS names that might be sensitive information.

- To ensure that the Host header in incoming HTTP packets contains the name understood by the internal Web server.

  Using the example in Figure 3-8 on page 113, suppose that the internal Web server expects all HTTP or HTTPS requests to have the **Host** field set to `data.com`. When users send requests using the published DNS name `novell.com/path`, the **Host** field of the packets in those requests received by the Access Gateway is set to `novell.com`. The Access Gateway can be configured to rewrite this public name to the private name expected by the Web server by setting the **Web Server Host Name** option to `data.com`. Before the Access Gateway forwards packets to the Web server, the **Host** field is changed (rewritten) from `novell.com` to `data.com`. For information about configuring this option, see “Configuring Web Servers of a Proxy Service” on page 100.

The rewriter searches for URLs in the following HTML contexts. They must meet the following criteria to be rewritten:

Context	Criteria
HTTP Headers	Qualified URL references occurring within certain types of HTTP response headers such as Location and Content-Location are rewritten. The Location header is used to redirect the browser to where the resource can be found. The Content-Location header is used to provide an alternate location where the resource can be found.
JavaScript	Within JavaScript, absolute references are always evaluated for rewriting. Relative references (such as `index.html`) are not attempted. Absolute paths (such as `/docs/file.html`) are evaluated if the page is read from a path-based multi-homing Web server and the reference follows an HTML tag. For example, the string `href='/docs/file.html'` is rewritten if `/docs` is a multi-homing path that has been configured to be removed.

HTML Tags	URL references occurring within the following HTML tag attributes are evaluated for rewriting:
	`action` `archive` `background`
	`cite` `code` `codebase`
	`data` `dynscr` `filterLink`
	`href` `longdesc` `lowsrc`
	`o:WebQuerySourceHref` `onclick` `onmenuclick`
	`pluginspage` `src` `usemap`
	`usermapborderimage`

Context	Criteria
References	An absolute reference is a reference that has all the information needed to locate a resource, including the hostname, such as `http://internal.web.site.com/index.html`. The rewriter always attempts to rewrite absolute references. The rewriter attempts to rewrite an absolute path when it is the multi-homing path of a path-based multi-homing service. For example, `/docs/file1.html` is rewritten if `/docs` is a multi-homing path that has been configured to be removed. Relative references are not rewritten.
Query Strings	URL references contained within query strings can be configured for rewriting by enabling the **Rewrite Inbound Query String Data** option.
Post Data	URL references specified in Post Data can be configured for rewriting by enabling the **Rewrite Inbound Post Data** option.

Specifying DNS Names to Rewrite

The rewriter parses and searches the Web content that passes through the Access Gateway for URL references that qualify to be rewritten. URL references are rewritten when they meet the following conditions:

- URL references containing DNS names or IP addresses matching those in the Web server address list are rewritten with the **Published DNS Name**.
- URL references matching the **Web Server Host Name** are rewritten with the **Published DNS Name**.
- URL references matching entries in the **Additional DNS Name List** of the host are rewritten with the **Published DNS Name**. The **Web Server Host Name** does not need to be included in this list.
- The DNS names in the **Exclude DNS Name List** specify the names that the rewriter must skip and not rewrite.

NOTE: Excludes in the **Exclude DNS Name List** are processed first, then the includes in the **Additional DNS Name List**. If you put the same DNS name in both lists, the DNS name is rewritten.

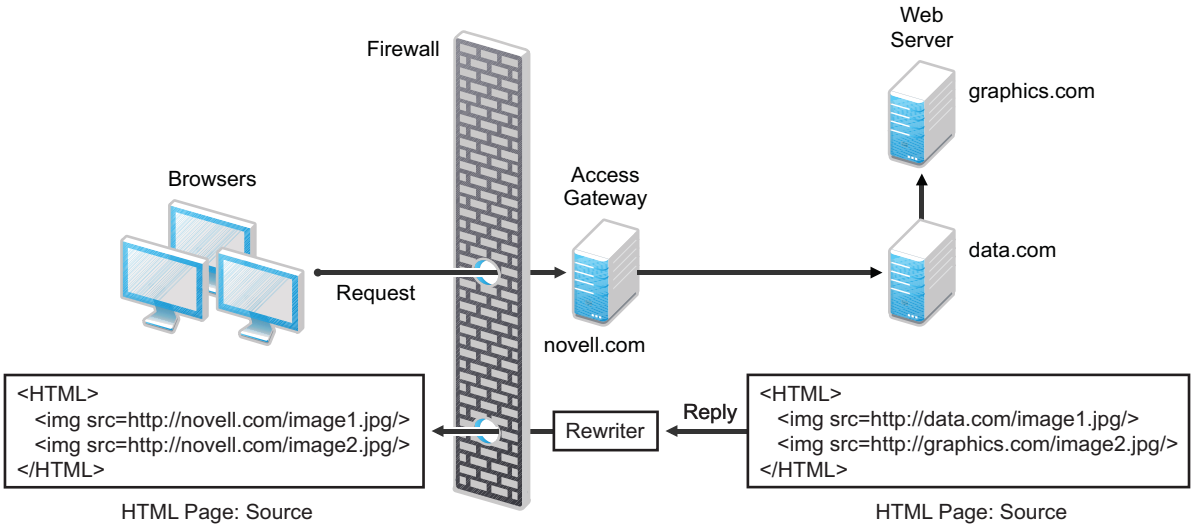
The following sections describe the conditions to consider when adding DNS names to the lists:

- “Determining Whether You Need to Specify Additional DNS Names” on page 116
- “Determining Whether You Need to Exclude DNS Names from Being Rewritten” on page 117

Determining Whether You Need to Specify Additional DNS Names

Sometimes Web pages contain URL references to a hostname that does not meet the default criteria for being rewritten. That is, the URL reference does not match the **Web Server Host Name** or any value (IP address) in the **Web Server List**. If these names are sent back to the client, they are not resolvable. Figure 3-9 illustrates a scenario that requires an entry in the **Additional DNS Name List**.

Figure 3-9 *Rewriting a URLs for Web Servers*



The page on the `data.com` Web server contains two links, one to an image on the `data.com` server and one to an image on the `graphics.com` server. The link to the `data.com` server is automatically rewritten to `novell.com`, when rewriting is enabled. The link to the image on `graphics.com` is not rewritten, until you add this URL to the **Additional DNS Name List**. When the link is rewritten, the browser knows how to request it, and the Access Gateway knows how to resolve it.

You need to include names in this list if your Web servers have the following configurations:

- If you have a cluster of Web servers that are not sharing the same DNS name, you need to add their DNS names to this list.
- If your Web server obtains content from another Web server, the DNS name for this additional Web server needs to be added to the list.
- If the Web server listens on one port (for example, 80), and redirects the request to a secure port (for example, 443), the DNS name needs to be added to the list. The response to the user comes back on `https://<DNS_name>:443`. This does not match the request that was sent on `http://<DNS_name>:80`. If you add the DNS name to the list, the response can be sent in the format that the user expects.
- If an application is written to use a private hostname, you need to add the private hostname to the list. For example, assume that an application URL reference contains the hostname of home (`http://home/index.html`). This hostname needs to be added to the **Additional DNS Name List**.
- If you enable the **Forward Received Host Name** option on your path-based multi-homing service and your Web server is configured to use a different port, you need to add the DNS name with the port to the **Additional DNS Name List**.

For example, if the public DNS name of the proxy service is `www.myag.com`, the path for the path-based multi-homing service is `/sales`, and the Web server port is 801, the following DNS name needs to be added to the **Additional DNS Name List** of the `/sales` service:

```
http://www.myag.com:801
```

When you enter a name in the list, it can use any of the following formats:

```
DNS_name
host_name
IP_address
scheme://DNS_name
scheme://IP_address
scheme://DNS_name:port
scheme://IP_address:port
```

For example:

```
HOME
https://www.backend.com
https://10.10.15.206:444
```

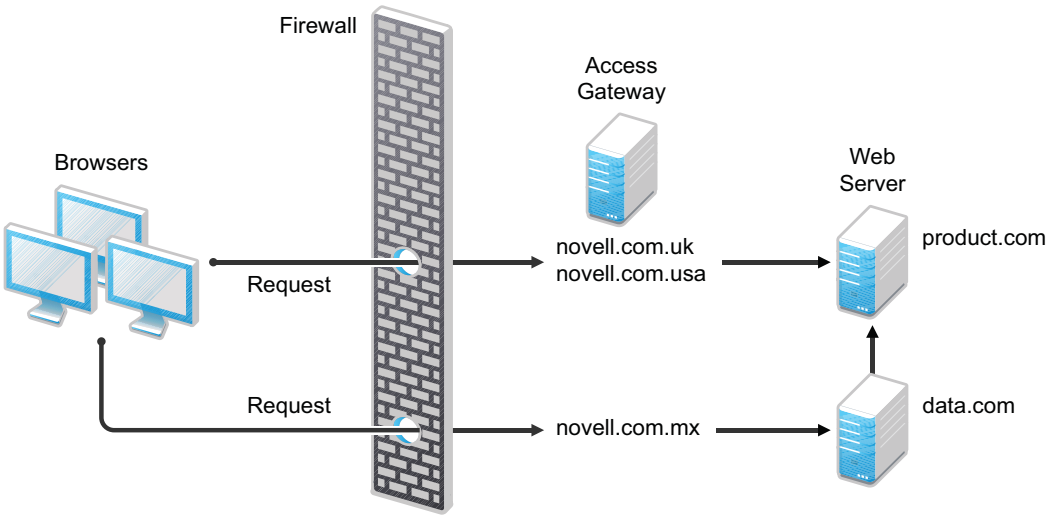
These entries are not case sensitive.

Determining Whether You Need to Exclude DNS Names from Being Rewritten

If you have two reverse proxies protecting the same Web server, the rewriter correctly rewrites the references to the Web server so that browser always uses the same reverse proxy. In other words, if the browser requests a resource using `novell.com.uk`, the response is returned with references to `novell.com.uk` and not `novell.com.usa`.

If you have a third reverse proxy protecting a Web server, the rewriting rules can become ambiguous. For example, consider the configuration illustrated in Figure 3-10.

Figure 3-10 *Excluding URLs*



A user accesses `data.com` through the published DNS name of `novell.com.mx`. The `data.com` server has references to product.com. The `novell.com.mx proxy` has two ways to get to the `product.com` server because this Web server has two published DNS names (`novell.com.uk` and `novell.com.usa`). The rewriter could use either of these names to rewrite references to `product.com`.

- If you want all users coming through `novell.com.mx` to use the `novell.com.usa` proxy, you need to block the rewriting of `product.com` to `novell.com.uk`. On the HTML Rewriting page of the reverse proxy for `novell.com.uk`, add `product.com` and any aliases to the **Exclude DNS Name List**.
- If you do not care which proxy is returned in the reference, you do not need to add anything to the **Exclude DNS Names List**.

Defining the Requirements for the Rewriter Profile

An HTML rewriter profile allows you to customize the rewriting process and specify the profile that is selected to rewrite content on a page. This section describes the following features of the rewriter profile:

- "Types of Rewriter Profiles" on page 118
- "Page Matching Criteria for Rewriter Profiles" on page 119
- "Possible Actions for Rewriter Profiles" on page 120
- "String Replacement Rules for Word Profiles" on page 122
- "String Tokens" on page 122
- "String Replacement Rules for Character Profiles" on page 123
- "Using $path to Rewrite Paths in JavaScript Methods or Variables" on page 124

Types of Rewriter Profiles

The Access Gateway has the following types of profiles:

- Default Word Profile
- Custom Word Profile
- Custom Character Profile

Default Word Profile

The default Word profile, named `default`, is not specific to a reverse proxy or its proxy services.

If you enable HTML rewriting, but you do not define a custom Word profile for the proxy service, the `default` Word profile is used. This profile is preconfigured to rewrite the **Web Server Host Name** and any other names listed in the **Additional DNS Name List**. The preconfigured profile matches all URLs with the following content-types:

text/html	text/javascript
text/xml	application/javascript
text/css	application/x-javascript

When you modify the behavior of the default profile, remember its scope. If the default profile does not match your requirements, you must usually create your own custom Word profile or custom Character profile.

Custom Word Profile

A Word profile searches for matches on words. For example, “get” matches the word “get” and any word that begins with “get” such as “getaway” but it does not match the “get” in “together” or “beget.”

For information about how strings are replaced in a Word profile, see the following:

- “String Replacement Rules for Word Profiles” on page 122
- “Using $path to Rewrite Paths in JavaScript Methods or Variables” on page 124

You must create a custom Word profile when an application requires rewrites of paths in JavaScript. If the application needs strings replaced or new content-types, these can also be added to the custom profile. In a custom Word profile, you can also configure the match criteria so that the profile matches specific URLs. For more information, see “Page Matching Criteria for Rewriter Profiles” on page 119.

When you create a custom Word profile, you need to position it before the `default` profile in the list of profiles. Only one Word profile is applied per page. The first Word profile that matches the page is applied. Profiles lower in the list are ignored.

Custom Character Profile

A custom Character profile searches for matches on a specified set of characters. For example, “top” matches the word “top” and the “top” in “tabletop,” “stopwatch,” and “topic.” If you need to replace strings that require this type of search, you must create a custom Character profile.

For information about how strings are replaced in a Character profile, see “String Replacement Rules for Character Profiles” on page 123.

In a custom Character profile, you can also configure the match criteria so that the profile matches specific URLs. For more information, see “Page Matching Criteria for Rewriter Profiles” on page 119.

After the rewriter finds and applies the Word profile that matches the page, it finds and applies one Character profile. The first Character profile that matches the page is applied. Character profiles lower in the list are ignored.

Page Matching Criteria for Rewriter Profiles

You specify the following matching criteria for selecting the profile:

- The URLs to match
- The URLs that cannot match
- The content types to match

You use the **Requested URLs to Search** section of the profile to set up the matching policy. The first Word profile and the first Character profile that matches the page is applied. Profiles lower in the list are ignored.

URLs: The URLs specified in the policy must use the following formats:

Sample URL	Description
`http://www.a.com/content`	Matches pages only if the requested URL does not contain a trailing slash.
`http://www.a.com/content/`	Matches pages only if the requested URL does contain a trailing slash.
`http://www.a.com/content/index.html`	Matches only this specific file.
`http://www.a.com/content/*`	Matches the requested URL whether or not it has a trailing slash and matches all files in the directory.
`http://www.a.com/*`	Matches the proxy service and everything it is protecting.

You can specify two types of URLs. In the **If Requested URL Is** list, you specify the URLs of the pages you want this profile to match. In the **And Requested URL Is Not** list, you specify the URLs you don't want this profile to match. You can use the asterisk wildcard for a URL in the **If Requested URL Is** list to match pages you really don't want this profile to match, then use a URL in the **And Requested URL Is Not** list to exclude them from matching. If a page matches both a URL in the **If Requested URL Is** list and in the **And Requested URL Is Not** list, the profile does not match the page.

For example, you could specify the following URL in the **If Requested URL Is** list:

`http://www.a.com/*`

You could then specify the following URL in the **And Requested URL Is Not** list:

`http://www.a.com/content/*`

These two entries cause the profile to match all pages on the `www.a.com` Web server except for the pages in the `/content` directory and its subdirectories.

IMPORTANT: If nothing is specified in either of the two lists, the profile skips the URL matching requirements and uses the content-type to determine if a page matches.

Content-Type: In the **And Document Content-Type Is** section, you specify the content-types you want this profile to match. To add a new content-type, click **New** and specify the name, such as `text/dns`. Search your Web pages for content-types to determine if you need to add new types. To add multiple values, enter each value on a separate line.

Regardless of content-types you specify, the page matches the profile if the file extension is `html`, `htm`, `shtml`, `jhtml`, `asp`, or `jsp` and you have not specified any URL matching criteria.

Possible Actions for Rewriter Profiles

The rewriter action section of the profile determines the actions the rewriter performs when a page matches the profile. Select from the following:

- Inbound Actions
- Enabling or Disabling Rewriting
- Additional Names to Search for URL Strings to Rewrite with Host Name
- String Replacement

Inbound Actions: A profile might require these options if the proxy service has the following characteristics:

- URLs appear in query strings, Post Data, or headers.
- The Web server uses WebDAV methods.

If your profile needs to match pages from this type of proxy service, you might need to enable the options listed below. They control the rewriting of query strings, Post Data, and headers from the Access Gateway to the Web server.

- **Rewrite Inbound Query String Data:** Select this option to rewrite the domain and URL in the query string to match the Web server configuration or to remove the path from the query string on a path-based multi-homing proxy with the **Remove Path on Fill** option enabled.
- **Rewrite Inbound Post Data:** Select this option to rewrite the domain and URL in the Post Data to match the Web server configuration or to remove the path from the Post Data on a path-based multi-homing proxy with the **Remove Path on Fill** option enabled.
- **Rewrite Inbound Headers:** Select this option to rewrite the following headers:

  Call-Back
  Destination
  If
  Notification-Type
  Referer

The inbound options are not available for a Character profile.

Enabling or Disabling Rewriting: The **Enable Rewriter Actions** option determines whether the rewriter performs any actions:

- Select the option to have the rewriter rewrite the references and data on the page.
- Leave the option deselected to disable rewriting. This allows you to create a profile for the pages you do not want rewritten.

Additional Names to Search for URL Strings to Rewrite with Host Name: Use this section to specify the name of the variable, attribute, or method in which the hostname might appear. These options are not available for a Character profile.

- **Variable and Attribute Name to Search for Is:** Use this section to specify the HTML attributes or JavaScript variables that you want searched for DNS names that might need to be rewritten. For the list of HTML attribute names that are automatically searched, see "HTML Tags" on page 114. You might want to add the following attributes:
  - **value:** This attribute enables the rewriter to search the `<param>` elements on the HTML page for value attributes and rewrite the value attributes that are URL strings.

    If you need more granular control (some need to be rewritten but others do not) and you can modify the page, see "Disabling with Page Modifications" on page 129.
  - **formvalue:** This attribute enables the rewriter to search the `<form>` element on the HTML page for `<input>`, `<button>`, and `<option>` elements and rewrite the value attributes that are URL strings. For example, if your multi-homing path is `/test` and the form line is `<input name="navUrl" type="hidden" value="/IDM/portal/cn/GuestContainerPage/656gwmail">`, this line would be rewritten to the following value before sending the response to the client:

```
<input name="navUrl" type="hidden" value="/test/IDM/portal/cn/
GuestContainerPage/656gwmail">
```

The formvalue attribute enables the rewriting of all URLs in the `<input>`, `<button>`, and `<option>` elements in the form.If you need more granular control (some need to be rewritten but others do not) and you can modify the form page, see "Disabling with Page Modifications" on page 129.

- **Replacing URLs in Java Methods:** The **JavaScript Method to Search for Is** list allows you to specify the Java methods to search to see if their parameters contain a URL string.

String Replacement: The **Additional Strings to Replace** list allows you to search for a string and replace it. The search boundary (word or character) that you specified when creating the profile is used when searching for the string.

Word profile search and replace actions take precedence over character profile actions.

For the rules and tokens that can be used in the search strings, see the following:

- "String Replacement Rules for Word Profiles" on page 122
- "String Tokens" on page 122
- "String Replacement Rules for Character Profiles" on page 123

For information about how the **Additional Strings to Replace** list can be used to reduce the number of Java methods you need to list, see "Using $path to Rewrite Paths in JavaScript Methods or Variables" on page 124.

String Replacement Rules for Word Profiles

In a Word profile, a string matches all paths that start with the characters in the specified string. For example:

Search String	Matches This String	Does not' Match This String
`/path`	`/path`	`/mypath`
	`/pathother`	
	`/path/other`	
	`/path.html`	

String Tokens

On the Access Gateway Service, you can use the following special tokens to modify the default matching rules.

- [w] to match one white space character
- [ow] to match 0 or more white space characters
- [ep] to match a path element in a URL path, excluding words that end in a period
- [ew] to match a word element in a URL path, including words that end in a period
- [oa] to match one or more alphanumeric characters

White Space Tokens: You use the [w] and the [ow] tokens to specify where white space might occur in the string. For example:

```
[ow]my[w]string[w]to[w]replace[ow]
```

If you don’t know, or don’t care, whether the string has zero or more white characters at the beginning and at the end, use [ow] to specify this. The [w] specifies exactly one white character.

Path Tokens: You use the [ep] and [ew] tokens to match path strings. The [ep] token can be used to match the following types of paths:

Search String	Matches This String	Does not’ Match This String
`/path[ep]`	`/path`	`/path.html`
	`/home/path/other`	`/home/pathother`

The [ew] token can be used to match the following types of paths:

Search String	Matches This String	Does not Match This String
`/path[ew]`	`/path.html`	`/paths`
	`/home/path`	

Name Tokens: You use the [oa] token to match function or parameter names that have a set string to start the name and end the name, but the middle part of the name is a computer-generated alphanumeric string. For example, the [oa] token can be used to match the following types of names:

Search String	Matches This String	Does not’ Match This String
`javaFunction-[oa](`	`javaFunction-1234a56()`	`javaFunction()`
	`javaFunction-a()`	

String Replacement Rules for Character Profiles

When you configure multiple strings for replacement, the rewriter uses the following rules for determining how characters are replaced in strings:

- String replacement is done as a single pass.
- String replacement is not performed recursively. Suppose you have listed the following search and replacement strings:

  ```
  DOG     to be replaced with     CAT
  A       to be replaced with     O
  ```

  All occurrences of the string DOG are replaced with CAT, regardless of whether it is the word DOG or the word DOGMA. Only one replacement pass occurs. The rewritten CAT is not replaced with COT.
- Because string replacement is done in one pass, the string that matches first takes precedence. Suppose you have listed the following search and replacement strings:

  ```
  ABC       to be replaced with     XYZ
  BCDEF     to be replaced with     PQRSTUVWXYZ
  ```

  If the original string is ABCDEFGH, the replaced string is XYZDEFGH.
- If two specified search strings match the data portion, the search string of longer length is used for the replacement except for the case detailed above. Suppose you have listed the following search and replacement strings:

```
ABC         to be replaced with     XYZ
ABCDEF      to be replaced with     PQRSTUVWXYZ
```

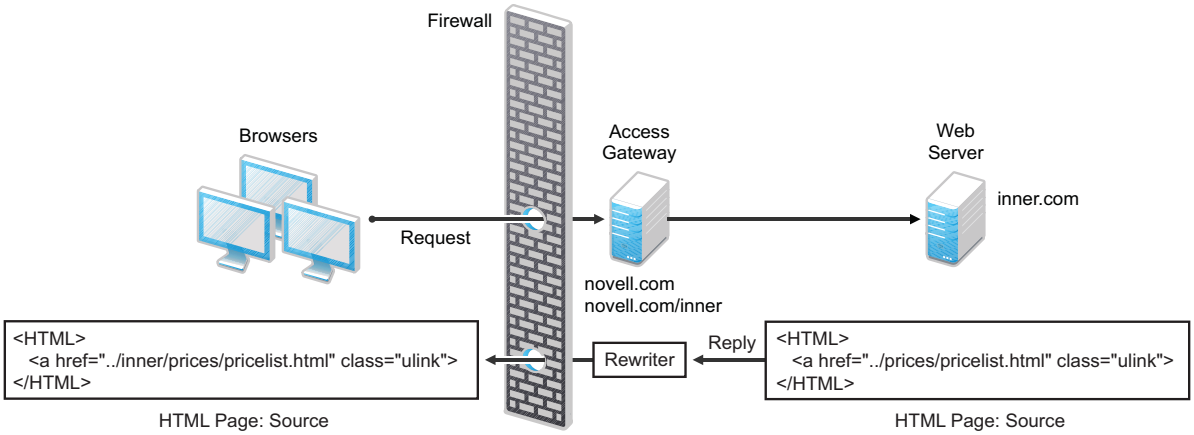
If the original string is ABCDEFGH, the replaced string is PQRSTUVWXYZGH.

Using $path to Rewrite Paths in JavaScript Methods or Variables

You can use the $path token to rewrite paths on a path-based multi-homing service that has the **Remove Path on Fill** option enabled. This token is useful for Web applications that require a dedicated Web server and are therefore installed in the root directory of the Web server. If you protect this type of application with Access Manager using a path-based multi-homing service, your clients access the application with a URL that contains a `/path` value. The proxy service uses the path to determine which Web server a request is sent to, and the path must be removed from the URL before sending the request to the Web server.

The application responds to the requests. If it uses JavaScript methods or variables to generate paths to resources, these paths are sent to client without prepending the path for the proxy service. When the client tries to access the resource specified by the Web server path, the proxy service cannot locate the resource because the multi-homing path is missing. The figure below illustrates this flow with the rewriter adding the multi-homing path in the reply.

Figure 3-11 *Rewriting with a Multi-homing Path*



To make sure all the paths generated by JavaScript are rewritten, you must search the Web pages of the application. You can then either list all the JavaScript methods and variables in the **Additional Names to Search for URL Strings to Rewrite with Host Name** section of the rewriter profile, or you can use the $path token in the **Additional Strings to Replace** section. The $path token reduces the number of JavaScript methods and variables that you otherwise need to list individually.

To use the $path token, you add a search string and a replace string that uses the token. For example, if the `/prices/pricelist.html` page is generated by JavaScript and the multi-homing path for the proxy service is `/inner`, you would specify the following stings:

Search String	Replacement String
`/prices`	`$path/prices`

This configuration allows the following paths to be rewritten before the Web server sends the information to the browser.

Web Server String	Rewritten String for the Browser
/prices/pricelist.html	/inner/prices/pricelist.html
/prices	/inner/prices

This token can cause strings that shouldn't be changed to be rewritten. If you enable the **Rewrite Inbound Query String Data**, **Rewrite Inbound Post Data,** and **Rewrite Inbound Header** actions, the rewriter checks these strings and ensures that they contain the information the Web server expects. For example, when these options are enabled, the following paths and domain names are rewritten when found in query strings, in Post Data, or in the Call-Back, Destination, If, Notification-Type, or Referer headers.

Browser String	Rewritten String for the Web Server
/inner/prices/pricelist.html	/prices/pricelist.html
/inner/prices	/prices
novell.com/inner/prices	inner.com/prices

Configuring the HTML Rewriter and Profile

You configure the HTML rewriter for a proxy service, and these values are applied to all Web servers that are protected by this proxy service.

To configure the HTML rewriter:

1. In the Administration Console, click **Devices** > **Access Gateways** > **Edit** > **[Name of Reverse Proxy]** > **[Name of Proxy Service]** > **HTML Rewriting**.

   The HTML Rewriting page specifies which DNS names are to be rewritten. The HTML Rewriter Profile specifies which pages to search for DNS names that need to be rewritten.

2. Select **Enable HTML Rewriting**.

   This option is enabled by default. When it is disabled, no rewriting occurs.When enabled, this option activates the internal HTML rewriter. This rewriter replaces the name of the Web server with the published DNS name when sending data to the browsers. It replaces the published DNS name with the **Web Server Host Name** when sending data to the Web server. It also makes sure the proper scheme (HTTP or HTTPS) is included in the URL. This is needed because you can configure the Access Gateway to use HTTPS between itself and client browsers and to use HTTP between itself and the Web servers.

3. In the **Additional DNS Name List** section, click **New**, specify a DNS that appears on the Web pages of your server (for example a DNS name other than the Web server's DNS name), then click **OK**.

   For more information, see "Determining Whether You Need to Specify Additional DNS Names" on page 116.

4. In the **Exclude DNS Name List** section, click **New**, specify a DNS name that appears on the Web pages of your server that you do not want rewritten, then click **OK**.

   For more information, see "Determining Whether You Need to Exclude DNS Names from Being Rewritten" on page 117.

5 Use the **HTML Rewriter Profile List** to configure a profile. Select one of the following actions:

- **New:** To create a profile, click **New**. Specify a display name for the profile and select either a **Word** or **Character** for the **Search Boundary**. Continue with "Creating or Modifying a Rewriter Profile" on page 127.
  - **Word:** A Word profile searches for matches on words. For example, "get" matches the word "get" and any word that begins with "get" such as "getaway" but it does not match the "get" in "together" or "beget."

    If you create multiple Word profiles, order is important. The first Word profile that matches the page is applied. Word profiles lower in the list are ignored.
  - **Character:** A Character profile searches for matches on a specified set of characters. For example, "top" matches the word "top" and the "top" in "tabletop," "stopwatch," and "topic."

    If you want to add functionality to the `default` profile, create a Character profile. It has all the functionality of a Word profile, except searching for attribute names and Java variables and methods. If you create multiple Character profiles, order is important. The first Character profile that matches the page is applied. Character profiles lower in the list are ignored.
- **Delete:** To delete a profile, select the profile, then click **Delete**.
- **Enable:** To enable a profile, select the profile, then click **Enable**.
- **Disable:** To disable a profile, select the profile, then click **Disable**.
- **Modify:** To view or modify the current configuration for a profile, click the name of the profile. Continue with "Creating or Modifying a Rewriter Profile" on page 127.

  The default profile is designed to be applied to all pages protected by the Access Gateway. It is not specific to a reverse proxy or its proxy services. If you modify its behavior, remember its scope. Rather than modify the default profile, you must create your own custom Word profile and enable it.

6 If you have more than one profile in the **HTML Rewriter Profile List**, use the up-arrow and down-arrow buttons to order the profiles.

If you create more than one profile, order becomes important. For example if you want to rewrite all pages with a general rewriter profile (with a URL such as `/*`) and one specific set of pages with another rewriter profile (with a URL such as `/doc/100506/*`), you need to have the specific rewriter profile listed before the general rewriter profile.

Even if multiple Word or Character profiles are enabled, a maximum of one Word profile and one Character profile is executed per page. The first Word profile and Character profile in the list that matches a page are executed, and the others are ignored.

7 Enable the profiles you want to use for this protected resource. Select the profile, then click **Enable**.

The `default` profile cannot be disabled. However, it is not executed if you have enabled another Word profile that matches your pages, and this profile comes before the `default` profile in the list.

8 To save your changes to browser cache, click **OK**.

9 To apply your changes, click the **Access Gateways** link, then click **Update** > **OK**.

10 The cached pages affected by the rewriter changes must be updated on the Access Gateway. Do one of the following:

- If the changes affect numerous pages, click **Access Gateways**, select the name of the server, then click **Actions** > **Purge All Cache**.
- If the changes affect only a few pages, you can refresh or reload the pages within the browser.

Creating or Modifying a Rewriter Profile

1 In the Administration Console, click **Devices** > **Access Gateways** > **Edit** > **[Name of Reverse Proxy]** > **[Name of Proxy Service]** > **HTML Rewriting**.

2 Select one of the following:

- To create a new profile, click **New**, specify a name, select a profile type, then click **OK**.
- To modify a profile, click the name of the profile.

3 Use the **Requested URLs to Search** section to set up a policy for specifying the URLs you want this profile to match.

Fill in the following fields:

If Requested URL Is: Specify the URLs of the pages you want this profile to match. Click **New** to add a URL to the text box. To add multiple values, enter each value on a separate line.

And Requested URL Is Not: Specify the URLs of pages that this profile must not match. If a page matches the URL in both the **If Requested URL Is** list and **And Requested URL Is Not** list, the profile does not match the page. Click **New** to add a URL to the text box. To add multiple values, enter each value on a separate line.

And Document Content-Type Is: Select the content-types you want this profile to match. To add a new content-type, click **New** and specify the name such as `text/dns`. Search your Web pages for content-types to determine if you need to add new types. To add multiple values, enter each value on a separate line.

For more information about how to use these options, see "Page Matching Criteria for Rewriter Profiles" on page 119.

4 Use the **Actions** section to specify the actions the rewriter must perform if the page matches the criteria in the **Requested URLs to Search** section.

Configure the following actions:

Rewrite Inbound Query String Data: (Not available for Character profiles) Select this option to rewrite the domain and URL in the query string to match the Web server. To use this option, your proxy service must meet the conditions listed in "Possible Actions for Rewriter Profiles" on page 120.

Rewrite Inbound Post Data: (Not available for Character profiles) Select this option to rewrite the domain and URL in the Post Data to match the Web server. To use this option, your proxy service must meet the conditions listed in "Possible Actions for Rewriter Profiles" on page 120.

Rewrite Inbound Headers: Select this option to rewrite the following headers:

Call-Back
Destination
If
Notification-Type
Referer

Enable Rewriter Actions: Select this action to enable the rewriter to perform any actions:

- Select it to have the rewriter use the profile to rewrite references and data on the page. If this option is not selected, you cannot configure the action options.
- Leave it unselected to disable rewriting. This allows you to create a profile for the pages you do not want rewritten.

5 (Not available for Character profiles) If your pages contain JavaScript, use the **Additional Names to Search for URL Strings to Rewrite with Host Name** section to specify JavaScript variables or methods. You can also add HTML attribute names. (For the list of attribute names that are automatically searched, see "HTML Tags" on page 114.)

Fill in the following fields:

Variable or Attribute Name to Search for Is: Lists the name of an HTML attribute or JavaScript variable to search to see if its value contains a URL string. Click **New** to add a name to the text box. To add multiple values, enter each value on a separate line.

JavaScript Method to Search for Is: Lists the names of Java methods to search to see if their parameters contain a URL string. Click **New** to add a method to the text box. To add multiple values, enter each value on a separate line.

6 Use the **Additional Strings to Replace** section to specify a string to search for and specify the text it must be replaced with. The search boundary (word or character) that you specified when creating the profile is used when searching for the string.

To add a string, click **New**, then fill in the following:

Search: Specify the string you want to search for. The profile type controls the matching and replacement rules. For more information, see one of the following:

- “String Replacement Rules for Character Profiles” on page 123
- “String Replacement Rules for Word Profiles” on page 122
- “Using $path to Rewrite Paths in JavaScript Methods or Variables” on page 124

Replace With: Specify the string you want to use in place of the search string.

7 Click **OK**.

8 If you have more than one profile in the **HTML Rewriter Profile List**, use the up-arrow and down-arrow buttons to order the profiles.

If you create more than one profile, order becomes important. For example if you want to rewrite all pages with a general rewriter profile (with a URL such as /*) and one specific set of pages with another rewriter profile (with a URL such as /doc/100506/*), you need to have the specific rewriter profile listed before the general rewriter profile.

Even if multiple Word or Character profiles are enabled, a maximum of one Word profile and one Character profile is executed per page. The first Word profile and Character profile in the list that matches a page are executed, and the others are ignored.

9 Enable the profiles you want to use for this protected resource. Select the profile, then click **Enable**.

The default profile cannot be disabled. However, it is not executed if you have enabled another Word profile that matches your pages, and this profile comes before the default profile in the list.

10 To save your changes to browser cache, click **OK**.

11 To apply your changes, click the **Access Gateways** link, then click **Update** > **OK**.

12 The cached pages affected by the rewriter changes must be updated on the Access Gateway. Do one of the following:

- If the changes affect numerous pages, click **Access Gateways**, select the name of the server, then click **Actions** > **Purge All Cache**.
- If the changes affect only a few pages, refresh or reload the page within the browser.

Disabling the Rewriter

There are three methods you can use to disable the internal rewriter:

- “Disabling per Proxy Service” on page 129
- “Disabling per URL” on page 129
- “Disabling with Page Modifications” on page 129

Disabling per Proxy Service

By default, the rewriter is enabled for all proxy services. The rewriter can slow performance because of the parsing overhead. In some cases, a Web site might not have content with URL references that need to be rewritten. The rewriter can be disabled on the proxy service that protects that Web site.

1. In the Administration Console, click **Devices** > **Access Gateways** > **Edit** > **[Name of Reverse Proxy]** > **[Name of Proxy Service]** > **HTML Rewriting**.
2. Deselect the **Enable HTML Rewriting** option, then click **OK**.
3. To apply your changes, click the **Access Gateways** link, then click **Update** > **OK**.
4. Select the Access Gateway, then click **Actions** > **Purge All Cache** > **OK**.

Disabling per URL

You can also specify a list of URLs that are to be excluded from being rewritten for the selected proxy service.

1. In the Administration Console, click **Devices** > **Access Gateways** > **Edit** > **[Name of Reverse Proxy]** > **[Name of Proxy Service]** > **HTML Rewriting**.
2. Click the name of the Word profile defined for this proxy service.

   If you have not defined a custom Word profile for the proxy service, you might want to create one. If you modify the `default` profile, those changes are applied to all proxy services.
3. In the **And Requested URL Is Not** section, click **New**, then specify the names of the URLs you do not want rewritten.

   Specify each URL on a separate line.
4. Click **OK** twice.
5. In the **HTML Rewriter Profile List**, make sure the profile you have modified is enabled and at the top of the list, then click **OK**.
6. To apply your changes, click the **Access Gateways** link, then click **Update** > **OK**.
7. Select the Access Gateway, then click **Actions** > **Purge All Cache** > **OK**.

Disabling with Page Modifications

There are cases when the URLs in only part of a page or in some of the JavaScript or form can be rewritten and the rest must not be rewritten. When this is the case, you might need to modify the content on the Web server. Although this deviates from the design behind Access Manager, you might encounter circumstances where it cannot be avoided.

You can add the following types of tags to the pages on the Web server:

- Page Tags
- Param Tags
- Form Tags

These tags are seen by browsers as a comment mark, and do not show up on the screen (except possibly on older browser versions).

NOTE: If the pages you modify are cached on the Access Gateway, you need to purge the cache before the changes become effective. Click **Access Gateways**, select the name of the server, then click **Actions** > **Purge All Cache**

Page Tags: If you want only portions of a page rewritten, you can add the following tags to the page.

```
<!--NOVELL_REWRITER_OFF-->
.
.
HTML data not to be rewritten
.
.
<!--NOVELL_REWRITER_ON-->
```

The last tag is optional, and if omitted, it prevents the rest of the page from being rewritten after the `<!--NOVELL_REWRITER_OFF-->` tag is encountered.

Param Tags: Sometimes the JavaScript on the page contains `<param>` elements that contain a value attribute with a URL. You can enable global rewriting of this attribute by adding `value` to the list of variable and attribute names to search for. If you need more control because some URLs need to be rewritten but others cannot be rewritten, you can turn on and turn off the `value` rewriting by adding the following tags before and after the `<param>` element in the JavaScript.

```
<!--NOVELL_REWRITE_ATTRIBUTE_ON='value'-->
.
.
<param> elements to be rewritten
.
.
<!--NOVELL_REWRITE_ATTRIBUTE_OFF='value'-->
.
.
<param> elements that shouldn't be rewritten
```

Form Tags: Some applications have forms in which the `<input>`, `<button>`, and `<option>` elements contain a value attribute with a URL. You can enable global rewriting of these attributes by adding `formvalue` to the list of variable and attribute names to search for. If you need more control because some URLs need to be rewritten but others cannot be rewritten, you can turn on and turn off the `formvalue` rewriting by adding the following tags before and after the `<input>`, `<button>`, and `<option>` elements in the form.

```
<!--NOVELL_REWRITE_ATTRIBUTE_ON='formvalue'-->
.
.
<input>, <button>, and <option> elements to be rewritten
.
.
<!--NOVELL_REWRITE_ATTRIBUTE_OFF='formvalue'-->
.
.
<input>, <button>, and <option> elements that shouldn't be rewritten
```

Configuring Connection and Session Limits

The Access Gateway establishes connections with clients and with Web servers. For most networks, the default values for unresponsive connections and sessions provide adequate performance, but you can fine-tune the options for your network, its performance requirements, and your users:

- “Configuring TCP Listen Options for Clients” on page 131
- “Configuring TCP Connect Options for Web Servers” on page 132
- “Configuring Connection and Session Persistence” on page 132
- “Configuring Web Servers” on page 133

Authentication time limits for inactivity sessions are configured on the contract and enforced by the Identity Server. For information about how to configure this limit, see “Assigning a Timeout Per Protected Resource” on page 110.

Configuring TCP Listen Options for Clients

The TCP listen options allow you to control how idle and unresponsive browser connections are handled and to optimize these processes for your network. For most networks, the default values provide adequate performance. If your network is congested and slow, you might want to increase some of the limits.

1 In the Administration Console, click **Devices** > **Access Gateways > Edit > [Name of Reverse Proxy] > TCP Listen Options**.

2 Select **Enable Persistent Connections** to allow the Access Gateway to establish a persistent HTTP connection between the Access Gateway and the browser. Usually, HTTP connections service only one request and response sequence. A persistent connection allows multiple requests to be serviced before the connection is closed.

   This option is enabled by default.

3 Specify values for the **TCP Listen Options**:

   Keep Alive Interval: Determines when an idle connection is closed. If no application data is exchanged over a connection for this amount of time, the connection is closed. This value limits how long an idle persistent connection is kept open. This setting is a compromise between freeing resources to allow additional inbound connections, and keeping connections established so that new connections from the same device do not need to be re-established. The value can be set from 1 to 1440 seconds (24 minutes). The default is 300 seconds (5 minutes).

   Data Read Timeout: Determines when an unresponsive connection is closed. When exchanging data, if an expected response from the connected device is not received within this amount of time, the connection is closed. This value might need to be increased for slow or congested network links. The value can be set from 1 to 3600 seconds (1 hour). The default is 120 seconds (2 minutes).

4 To configure the encryption key, select one or more of the following:

   Enforce 128-Bit Encryption between Browser and Access Gateway: When this option is selected, the Access Gateway requires all its server connections with client browsers to use 128-bit encryption. If the encryption key is less than 128, regardless of the cipher suite, the connection is denied.

   Enforce 128-Bit Encryption between Access Gateway and Web Server: When this option is selected, the Access Gateway requires all its client connections to Web servers to use 128-bit encryption. If the encryption key is less than 128, regardless of the cipher suite, the connection is denied.

   NOTE: These SSL listening options appear disabled if you are configuring the tunneling services.

   ---

5 To save your changes to browser cache, click **OK**.

6 To apply your changes, click the **Access Gateways** link, then click **Update** > **OK**.

Configuring TCP Connect Options for Web Servers

Connect options are specific to the group of Web servers configured for a proxy service. They allow you to control how idle and unresponsive Web server connections are handled and to optimize these processes for your network. For most networks, the default values provide adequate performance. If your network is congested and slow, you might want to increase some of the limits.

1. In the Administration Console, click **Devices** > **Access Gateways** > **Edit** > **[Name of Reverse Proxy]** > **[Name of Proxy Service]** > **Web Servers** > **TCP Connect Options**.
2. Configure the IP address to use when establishing connections with Web servers:

   Cluster Member: (Available only if the Access Gateway is a member of a cluster.) Select the server you want to configure from the list of servers. Only the value of the **Make Outbound Connection Using** option applies to the selected server.
3. Select how the Web servers must be contacted when multiple Web servers are available. Select one of the following for the **Policy for Multiple Destination IP Addresses** option:
   - **Simple Failover:** Allows the next available Web server in the group to be contacted when the first server in the list is no longer available.
   - **Round Robin:** Moves in order through the list of Web servers, allowing each to service requests before starting at the beginning of the list for a second group of requests.
4. Select **Enable Persistent Connections** to allow the Access Gateway to establish a persistent HTTP connection between the Access Gateway and the Web server. Usually, HTTP connections service only one request and response sequence. A persistent connection allows multiple requests to be serviced before the connection is closed.

   This option is enabled by default.
5. To modify the connection timeouts between the Access Gateway and the Web servers, configure the following fields:

   Data Read Timeout: Determines when an unresponsive connection is closed. When exchanging data, if an expected response from the connected device is not received within this amount of time, the connection is closed. This value might need to be increased for slow or congested network links. The value can be set from 1 to 3600 seconds (1 hour). The default is 120 seconds (2 minutes).

   Idle Timeout: Determines when an idle connection is closed. If no application data is exchanged over a connection for this amount of time, the connection is closed. This value limits how long an idle persistent connection is kept open. This setting is a compromise between freeing resources to allow additional inbound connections, and keeping connections established so that new connections from the same device do not need to be re-established. The value can be set from 1 to 1800 seconds (30 minutes). The default is 180 seconds (3 minutes).
6. To save your changes to browser cache, click **OK**.
7. To apply your changes, click the **Access Gateways** link, then click **Update** > **OK**.

Configuring Connection and Session Persistence

The Access Gateway establishes three types of connections:

- Access Gateway to browser
- Access Gateway to Web server
- Browser to Web server

The Access Gateway connections to the browser and the Access Gateway connections to the Web server involve setting up a TCP connection for an HTTP request. HTTP connections usually service only one request and response sequence, and the TCP connection is opened and closed during the sequence. A persistent connection allows multiple requests to be serviced before the connection is closed and saves a significant amount of processing time. To configure this type of persistence, see the following:

- **Access Gateway to Browser:** Click **Devices** > **Access Gateways** > **Edit** > **[Name of Reverse Proxy]** > **TCP Listen Options** and configure the **Enable Persistent Connections** option.
- **Access Gateway to Web Server:** Click **Devices** > **Access Gateways** > **Edit** > **[Name of Reverse Proxy]** > **[Name of Proxy Service]** > **Web Servers** > **TCP Connect Options** and configure the **Enable Persistent Connections** option.

The persistence of the browser to Web server connection is always enabled and is not configurable. This feature allows a browser to use the same Web server after an initial connection has been established. Most Web applications are designed to expect this type of behavior.

Configuring Web Servers

The Web server configuration determines how the Access Gateway handles connections and packets between itself and the Web servers. For more information about Web Server configuration, see “Configuring Web Servers of a Proxy Service” on page 100

1. Click **Access Gateways** > **Edit** > **[Name of Reverse Proxy]** > **[Name of Proxy Service]** > **Web Servers**.
2. If your browsers are capable of sending HTTP 1.1 requests, configure the following field to match your Web servers:

   Enable Force HTTP 1.0 to Origin: Indicates whether HTTP 1.1 requests from browsers are translated to HTTP 1.0 requests before sending them to the Web server. If your browsers are sending HTTP 1.1 requests and your Web server can only handle HTTP 1.0 requests, you must enable this option.

   When the option is enabled, the Access Gateway translates an HTTP 1.1 request to an HTTP 1.0 request.
3. To enable SSL connections between the proxy service and its Web servers, select **Connect Using SSL**. For configuration information for this option, **Web Server Trusted Root**, and **SSL Mutual Certificate**, see “Configuring SSL between the Proxy Service and the Web Servers” on page 828.
4. In the **Connect Port** field, specify the port that the Access Gateway must use to communicate with the Web servers. The following table lists some default port values for common types of Web servers.

Server Type	Non-Secure Port	Secure Port
Web server with HTML content	80	443
WebSphere	9080	9443
JBoss	8080	8443

5. To control how idle and unresponsive Web server connections are handled and to optimize these processes for your network, select **TCP Connect Options**. For more information, see “Configuring TCP Connect Options for Web Servers” on page 132.

6 To add a Web server, click **New** in the **Web Server List** and specify the IP address or the fully qualified DNS name of the Web server.

- **New:** To create a new Web server, click **New**. Specify the Web Server IP Address or DNS. Click OK to add the new Web server to the list or Cancel to discard the changes.

  After creating the Web server in the list, you can configure it as primary server and prioritize the list of Web servers based on your requirement.

- **Delete:** To delete a Web server, select the Web server from the list, then click **Delete**.

  If you delete the selected Web server, then all the Web servers which are corresponding to the device in the cluster gets deleted.

7 To save your changes to browser cache, click **OK**.

8 To apply your changes, click the **Access Gateways** link, then click **Update** > **OK**.

Protecting Multiple Resources

This section describes how to create multiple resources for Access Gateways.

Using Multi-Homing to Access Multiple Resources

You can configure an Access Gateway to use one public IP address to protect multiple types of Web resources. This is one of the major benefits of the Access Gateway, because it conserves valuable resources such as IP addresses. This feature also makes an Access Gateway a multi-homing device because it becomes a single endpoint supporting multiple back-end resources.

You can select to use only one multi-homing method, or you can use multiple methods. Select the methods that meet the needs of your network and the resources you are protecting. The first proxy service configured for a reverse proxy is always configured to use the DNS name of the Access Gateway. Subsequent proxy services can be configured to use one of the following methods:

- “Domain-Based Multi-Homing” on page 134
- “Path-Based Multi-Homing” on page 136
- “Virtual Multi-Homing” on page 139

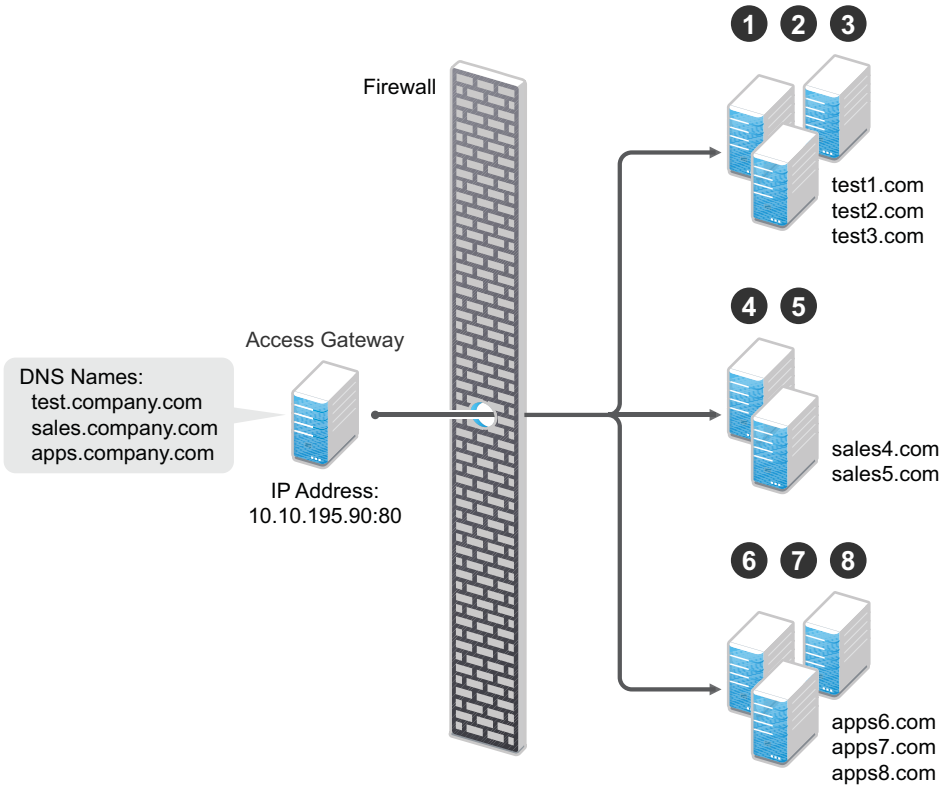
This section describes these multi-homing methods, then explains the following:

- “Creating a Second Proxy Service” on page 139
- “Configuring a Path-Based Multi-Homing Proxy Service” on page 140

Domain-Based Multi-Homing

Domain-based multi-homing is based on the cookie domain. For example, if you have a cookie domain of `company.com`, you can prefix hostnames to a cookie domain name. For a test resource, you can prefix `test` to `company.com` and have `test.company.com` resolve to the IP address of the Access Gateway. The Access Gateway configuration for the `test.company.com` proxy service contains the information for accessing its Web servers (`test1.com`). Figure 3-12 illustrates this type of configuration for three proxy services.

Figure 3-12 *Using a Base Domain Name with Host Names*



Domain-based multi-homing has the following characteristics:

- If you are using SSL, the back-end servers can all listen on the same SSL port (the default for HTTPS is 443).
- If you are using SSL, the back-end servers can share the same SSL certificate. Instead of using a specific hostname in the SSL certificate, the certificate can use a wildcard name such as `*.company.com`, which matches all the servers.

Before configuring the Access Gateway, you need to complete the following:

- Create the published DNS names with a common domain name for public access to the back-end resources. For example, the table below lists three DNS names that use `company.com` as a common domain name, lists the IP address that these DNS names resolve to, and the Web servers they protect.

Published DNS Name	Access Gateway IP Address	Web Server Host Name	Web Server IP Address
test.company.com	10.10.195.90:80	test.internal.com	10.10.15.10
sales.company.com	10.10.195.90:80	sales.internal.com	10.10.15.20
apps.company.com	10.10.195.90:80	apps.internal.com	10.10.15.30

- Configure your DNS server to resolve the published DNS names to the IP address of the Access Gateway.
- Set up the back-end Web servers.

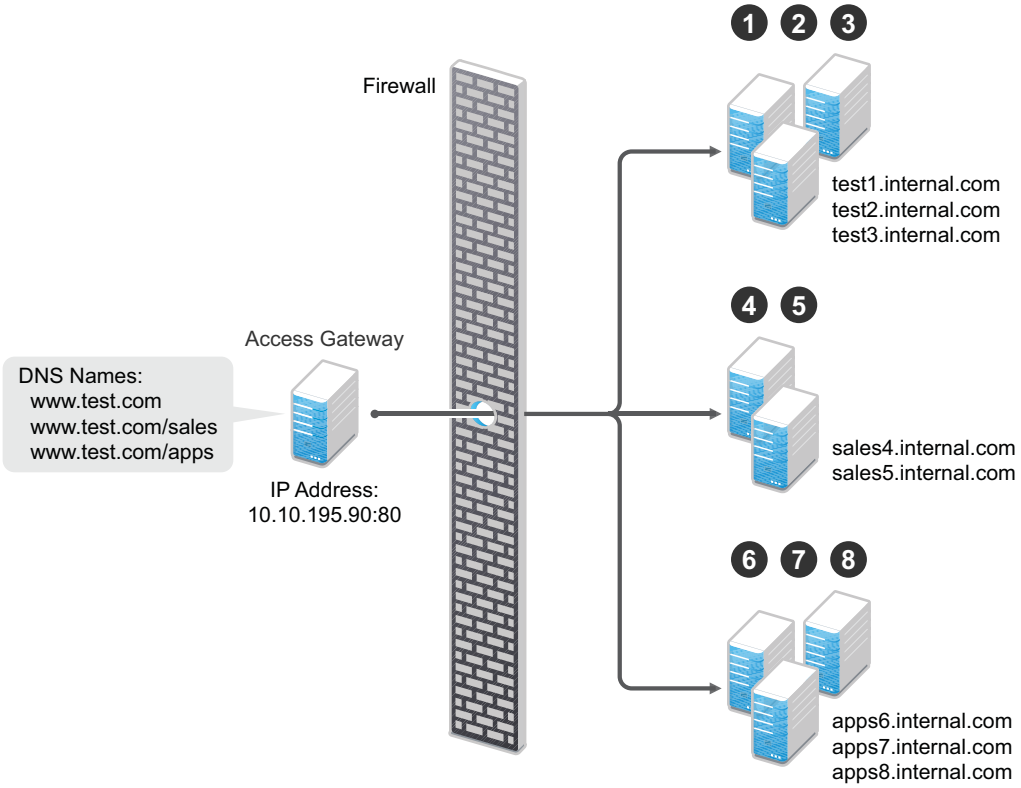
- Create three proxy services for these published DNS names.

  To create a domain-based multi-homing proxy service, see “Creating a Second Proxy Service” on page 139, and select domain-based for the multi-homing type.

Path-Based Multi-Homing

Path-based multi-homing uses the same DNS name for all resources, but each resource or resource group must have a unique path appended to the DNS name. For example, if the DNS name is `test.com`, you would append `/sales` to `test.com`. When the user enters the URL of `www.test.com/sales`, the Access Gateway resolves the URL to the sales resource group. Figure 3-13 illustrates this type of configuration.

Figure 3-13 *Using a Domain Name with Path Elements*



Path-based multi-homing has the following characteristics:

- It is considered to be more secure than domain-based multi-homing, because some security experts consider wildcard certificates less secure than a certificate with a specific hostname.
- Each resource or group of resources must have a unique starting path.
- JavaScript applications might not work as designed if they obscure the URL path. The Access Gateway needs access to the URL path, and if it is obscured, the path cannot be resolved to the correct back-end resource.
- The protected resources for each path-based child come from the parent proxy service.

The following sections explain how to configure path-based proxy services and your network so that the Access Gateway can find the correct protected resources:

- Configuring the Remove the Path on Fill Option
- Configuring the Host Header Option

- Configuring the Host Header Option
- Preparing for Path-Based Multi-Homing

Configuring the Remove the Path on Fill Option

If the path that is part of the published DNS name (`/sales` or `/apps`) is used to identify a resource but is not part of directory configuration on the Web server, the path needs to be removed from the URL before the request is sent to the Web server. For example, suppose you use the following configuration:

Browser URL Using the Published DNS Name	Web Server URL
http://www.test.com/sales	http://sales4.internal.com/

In this case, the path needs to be removed from the URL that the Access Gateway sends to the Web server. The Access Gateway does not allow you to set up multiple paths to this type of Web server, so all pages must have the same authentication requirements.

If the path in the published DNS name is a path on the Web server, the path needs to be passed to the Web server as part of the URL. For example, suppose you use the following configuration:

Browser URL Using the Published DNS Name	Web Server URL
http://www.test.com/sales	http://sales4.internal.com/sales

Because the path component specifies a directory on the Web server where the content begins, you need to select to include the path. The Access Gateway then includes the path as part of the URL it sends to the Web server. This configuration allows you to set up multiple paths to the Web server, such as

- sales/payroll
- sales/reports
- sales/products

Such a configuration also allows you to set up different authentication and authorization requirements for each path.

Configuring the Host Header Option

When you create path-based proxy services and also enable the **Remove Path on Fill** option, you need to know what types of links exist on the Web servers. For example, you need to know if the sales Web servers in Figure 3-13 on page 136 have links to the app Web servers or to the test Web servers. If they don't, you can set the **Host Header** option to either **Forward Received Host Name** or to **Web Server Host Name**. However, if they do contain links to each other, you need to set the **Host Header** option to **Web Server Host Name** and specify a DNS name for the Web server in the **Web Server Host Name** option. The Access Gateway needs a method to distinguish between the Web servers other than the path, because after the path is removed, all the Web servers in Figure 3-13 on page 136 have the same name: `www.test.com`.

If you select to use the **Forward Received Host Name** option for a path-based service, you might also need to add entries to the **Additional DNS Name List** for the rewriter. For more information, see "Determining Whether You Need to Specify Additional DNS Names" on page 116.

Preparing for Path-Based Multi-Homing

Before configuring the Access Gateway, you need to complete the following:

- Create the published DNS names with paths for public access to the back-end resources. For example, the table below uses `test.com` as the domain name. It lists three published DNS names (two with paths), the IP address these names resolve to, and the Web servers that they are going to protect:

Published DNS Name	Access Gateway IP Address	Web Server Host Name	Web Server IP Address
test.com	10.10.195.90:80	test.internal.com	10.10.15.10
test.com/sales	10.10.195.90:80	sales.internal.com	10.10.15.20
test.com/apps	10.10.195.90:80	apps.internal.com	10.10.15.30

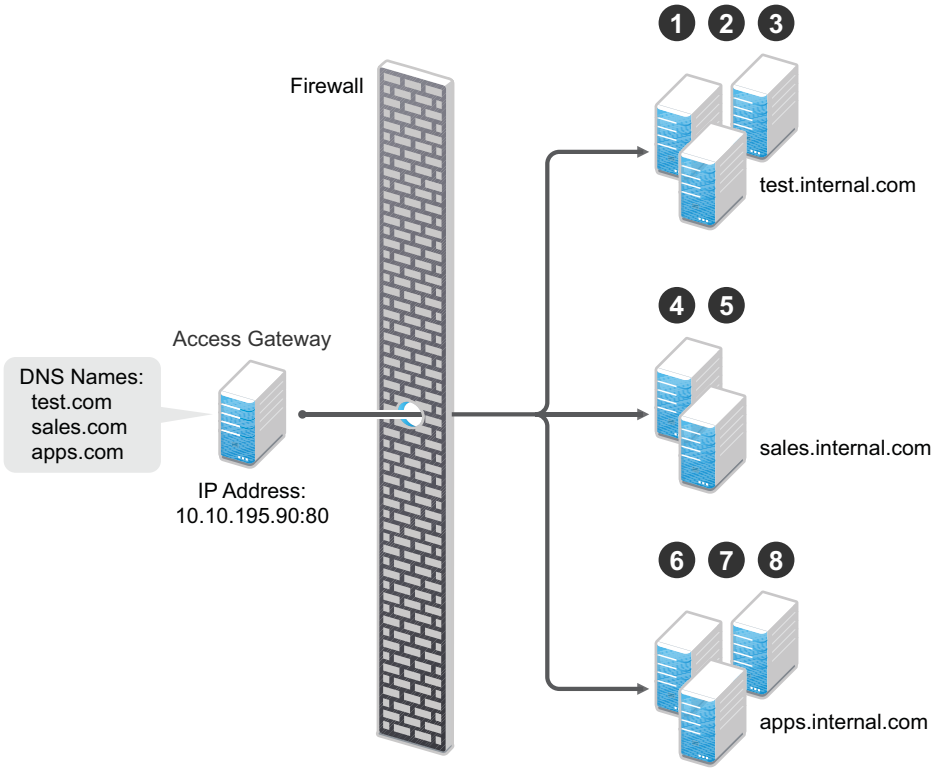
- Configure your DNS server to resolve the published DNS names to the IP address of the Access Gateway.
- Set up the back-end Web servers. If they have links to each other, set up DNS names for the Web servers.
- Create one proxy service that uses `test.com` as its published DNS name and two path-based proxy services.

  To create a path-based multi-homing proxy service, see “Creating a Second Proxy Service” on page 139, and select path-based for the multi-homing type.

Virtual Multi-Homing

Virtual multi-homing allows you to use DNS names from different domains (for example `test.com` and `sales.com`). Each of these domain names must resolve to the Access Gateway host. Figure 3-14 illustrates this type of configuration.

Figure 3-14 *Using Multiple DNS Names*



Virtual multi-homing cannot be used with SSL. You must use this configuration with resources that need to be protected, but the information exchanged must be public information that does not need to be secure. For example, you could use this configuration to protect your Web servers that contain the catalog of your shipping products. It isn't until the user selects to order a product that you need to switch the user to a secure site.

Whether a client can use one DNS name or multiple DNS names to access the Access Gateway depends upon the configuration of your DNS server. After you have configured your DNS server to allow multiple names to resolve to the same IP address, you are ready to configure the Access Gateway.

To create a virtual multi-homing proxy service, see "Creating a Second Proxy Service" on page 139, and select **Virtual** for the multi-homing type.

Creating a Second Proxy Service

1. In the Administration Console, click **Devices** > **Access Gateways** > **Edit** > **[Name of Reverse Proxy]**.
2. In the **Proxy Service List**, select **New**.
3. Fill in the fields.

   Proxy Service Name: Specify a display name for the proxy service. For the sales group, you might use sales. For the group of application servers, you might use apps.

Multi-Homing Type: Specify the multi-homing method that the Access Gateway must use to identify this proxy service. Select one of the following:

- **Domain-Based:** Uses the published DNS name (`www.test.com`) with a hostname (`www.newsite.test.com`). For more information, see "Domain-Based Multi-Homing" on page 134.
- **Path-Based:** Uses the published DNS name (`www.test.com`) with a path (`www.test.com/path`). For more information, see "Path-Based Multi-Homing" on page 136.
- **Virtual:** Uses a unique DNS name (`www.newsite.newcompany.com`). Virtual multi-homing cannot be used with SSL. For more information, see "Virtual Multi-Homing" on page 139. If you need a unique DNS name and SSL, you need to create a reverse proxy rather than a proxy service. For information about creating a second reverse proxy, see "Managing Multiple Reverse Proxies" on page 143.

Published DNS Name: Specify the DNS name you want the public to use to access your site. This DNS name must resolve to the IP address you set up as the listening address. This option is not available when path-based multi-homing is selected.

Path: Specify the path to use for this proxy service. This option is available only when path-based multi-homing is selected.

Web Server IP Address: Specify the IP address of the Web server you want this proxy service to manage.

Host Header: Specify whether the HTTP header must contain the name of the back-end Web server (**Web Server Host Name** option) or whether the HTTP header must contain the published DNS name (the **Forward Received Host Name** option).

For a path-based multi-homing service, it is usually best to select the **Web Server Host Name** option. For more information, see "Configuring the Host Header Option" on page 137.

Web Server Host Name: Specify the DNS name of the Web server that the Access Gateway must forward to the Web server. If you have set up a DNS name for the Web server and the Web server requires its DNS name in the HTTP header, specify that name in this field. If you selected **Forward Received Host Name**, this option is not available.

For iChain administrators, the **Web Server Host Name** is the alternate hostname when configuring a Web Server Accelerator.

4 Click **OK**.

5 To continue, select one of the following:

- To configure a virtual or domain-based proxy service, see "Configuring a Proxy Service" on page 97.
- To configure a path-based proxy service, see "Configuring a Path-Based Multi-Homing Proxy Service" on page 140.

Configuring a Path-Based Multi-Homing Proxy Service

To configure a path-based proxy service:

1 In the Administration Console, click **Devices** > **Access Gateways** > **Edit** > **[Name of Reverse Proxy]** > **[Name of Path-Based Multi-Homing Proxy Service]**.

The following fields display information that must be configured on the parent proxy service (the first proxy service created for this reverse proxy).

Published DNS Name: Displays the value that users are currently using to access this proxy service. This DNS name must resolve to the IP address you set up as a listening address on the Access Gateway.

Cookie Domain: Displays the domain for which the cookie is valid. The Web server that the user is accessing must be configured to be part of this domain.

2 Configure the following options:

Description: (Optional) Provide a description of the purpose of this proxy service or specify any other pertinent information.

HTTP Options: Determines how the proxy service handles HTTP headers and caching. For more information, see "Controlling Browser Caching" on page 254.

Advanced Options: (Access Gateway Service) See "Configuring the Advanced Options for a Domain-Based and Path-Based Multi-Homing Proxy Service" on page 269.

3 Configure the path options:

Remove Path on Fill: Determines whether the multi-homing path is removed from the URL before forwarding it to the Web server. If the path is not a directory at the root of the Web server, the path must be removed. If this option is selected, the path is stripped from the request before the request is sent to the Web server.

If you enable this option, this proxy service can protect only one path. If you have configured multiple paths in the **Path List**, you cannot enable this option until you have deleted all but one path.

Reinsert Path in "set-cookie" Header: Determines whether the path is inserted into the Set-Cookie header. This option is only available if you enable the **Remove Path on Fill** option.

4 Determine whether you need to create a protected resource for your path.

In the **Path List**, the path you specified is listed along with the protected resource that best matches its path.

The Access Gateway automatically selects the protected resource that is used with the specified path. It selects the current protected resource whose URL path most closely matches the specified path.

- If you have a protected resource with a URL path of /*, the Access Gateway selects that resource unless you have configured a protected resource that has a URL path that more closely matches the path specified on this page.
- If you add a protected resource at a future time and its URL path more closely matches the path specified on this page, the Access Gateway automatically reconfigures to use this new protected resource.
- If you disable a protected resource that the Access Gateway has assigned to a path-based service, the Access Gateway automatically reconfigures and selects the next protected resource that most closely matches the path specified on this page.

4a In the **Path List** section, click the **Protected Resource** link.

4b Examine the contract, Authorization, Identity Injection, and Form Fill policies assigned to this protected resource to ensure that they meet the requirements for your path-based service.

4c To return to the Path-Based Multi-Homing page, click the **Overview** tab, then click **OK**.

- If the protected resource meets your needs, continue with Step 5
- If the protected resource does not meet your needs, you must create a protected resource for the path-based proxy service. Continue with Step 4d.

4d Click **OK**, select the name of the parent proxy service, then click **Protected Resources**.

4e In the **Protected Resource List**, click **New**, specify a name, then click **OK**.

4f Select an Authentication Procedure.

4g In the **URL Path List**, specify the path you used when creating the path-based proxy service. For example, if your path was `/apps`, specify `/apps/*` or `/apps` in the URL Path List.

IMPORTANT: If you create multiple protected resources that exactly match the path-based multi-homing service, there is no guarantee that a specific protected resource will be used. For example, if you create protected resources for both of the paths specified above (`/apps` and `/apps/*`) and you have a path-based service with a path of `/apps`, either of these protected resources could be assigned to this path-based service in the Administration Console or used when access is requested.

4h Make sure the protected resource you created is enabled. If the resource is disabled, it does not appear in the Path List for the path-based proxy service.

4i (Optional) Enable the policies the path-based proxy service requires. Click **Authorization**, **Identity Injection**, or **Form Fill** and enable the appropriate policies.

4j Click **OK**.

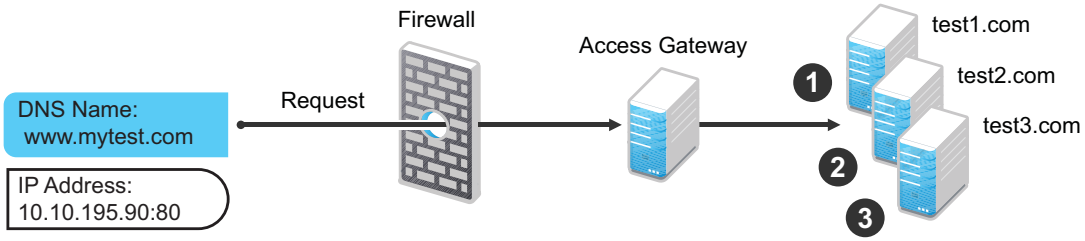
5 Click **OK**.

6 To apply the changes, click the **Access Gateways** link, then click **Update** > **OK**.

Setting Up a Group of Web Servers

You can configure a proxy service to service a “virtual” group of Web servers, which adds load balancing and redundancy. Each Web server in the group must contain the same material. When you create the proxy service, you set up the first server by specifying the URLs you want users to access and the rights the users need for each URL. When you add additional Web servers to the proxy service, these servers automatically inherit everything you have configured for the first Web server.

Figure 3-15 *Adding Redundant Web Servers*



For this configuration, you use a single reverse proxy and proxy service. To add multiple Web servers to a host:

1 In the Administration Console, click **Devices** > **Access Gateways** > **Edit** > **[Name of Reverse Proxy]** > **[Name of Proxy Service]** > **Web Servers**.

2 In the **Web Server List** section, click **New**.

3 Specify the IP address or the fully qualified DNS name of another Web server for the “virtual” group, then click **OK**.

4 Repeat Step2 and Step3 to add additional Web servers to the group.

5 Click **OK**.

6 To apply the changes, click the **Access Gateways** link, then click **Update** > **OK**.

The Access Gateway uses the round robin algorithm by default. You can configure it to use the simple failover algorithm.

Simple failover sends all the traffic to the first Web server as long as it is available. Traffic is sent to another Web server in the list only when the first Web server is no longer available. To configure this option, see “Configuring TCP Connect Options for Web Servers” on page 132.

Connection persistence is enabled by default. This allows the Access Gateway to send multiple HTTP requests to the Web server to be serviced before the connection is closed. To configure this option, see “Configuring TCP Connect Options for Web Servers” on page 132.

Session stickiness option is used if multiple Web Servers are configured for a service. Selecting this option makes the proxy server to use the same Web server for all fills during a session. This option is enabled by default. For more information about persistent connections, see “Configuring Connection and Session Persistence” on page 132.

Managing Multiple Reverse Proxies

Each reverse proxy must have a unique IP address and port combination. If your Access Gateway has only one IP address, you must select unique port numbers for each additional reverse proxy that you create. You can configure the Access Gateway to use multiple IP addresses. These addresses can be configured to use the same network interface card, or if you have installed multiple network cards, you can assign the IP addresses to different cards.

You need to use system utilities to configure network interface cards and new IP addresses. After they are configured, you can use the **New IP** option to make them available for Gateway Service configuration. See “Adding a New IP Address to the Access Gateway” on page 245.

If you are creating more than one reverse proxy, you must select one to be used for authentication. By default, the first reverse proxy you create is assigned this task. Depending upon your Access Gateway configuration, you might want to set up one reverse proxy specifically for handling authentication. The authentication reverse proxy is also used for logout. If you have Web applications that contain logout options, these options need to be redirected to the Logout URL of the authentication proxy.

- “Managing Entries in the Reverse Proxy List” on page 143

Managing Entries in the Reverse Proxy List

1. In the Administration Console, click **Devices** > **Access Gateways** > **Edit** > **Reverse Proxy / Authentication**.
2. In the **Reverse Proxy List**, select one of the following actions:
   - **New:** To create a new reverse proxy, click **New**. You are prompted to enter a display name for the proxy. For configuration information, see “Managing Reverse Proxies and Authentication” on page 94.

     Reverse proxy names and proxy service names must be unique to the Access Gateway. Protected resource names need to be unique to the proxy service, but they don’t need to be unique to the Access Gateway.
   - **Delete:** To delete a reverse proxy, select the check box next to a specific reverse proxy, then click **Delete**. To delete all reverse proxies, select the check box next to the **Name** column, then click **Delete**.
   - **Enable:** To enable a reverse proxy, select the check box next to a specific reverse proxy, then click **Enable**. To enable all reverse proxies, select the check box next to the **Name** column, then click **Enable**.
   - **Disable:** To disable a reverse proxy, select the check box next to a specific reverse proxy, then click **Disable**. To enable all reverse proxies, select the check box next to the **Name** column, then click **Disable**.

3 Click **OK**.

4 To apply the changes, click the **Access Gateways** link, then click **Update** > **OK**.

Configuring Trusted Providers for Single Sign-On

This section discusses configuring trust so that two user accounts can be associated with each other without the sites exchanging user data. Topics include:

- "Understanding the Trust Model" on page 144
- "Configuring General Provider Settings" on page 146
- "Managing Trusted Providers" on page 150
- "Modifying a Trusted Provider" on page 154
- "Communication Security" on page 155
- "Selecting Attributes for a Trusted Provider" on page 155
- "Managing Metadata" on page 158
- "Configuring an Authentication Response for a Service Provider" on page 162
- "Routing to an External Identity Provider Automatically" on page 162
- "Configuring Options for Trusted Service Providers" on page 163
- "Using the Intersite Transfer Service" on page 164

Understanding the Trust Model

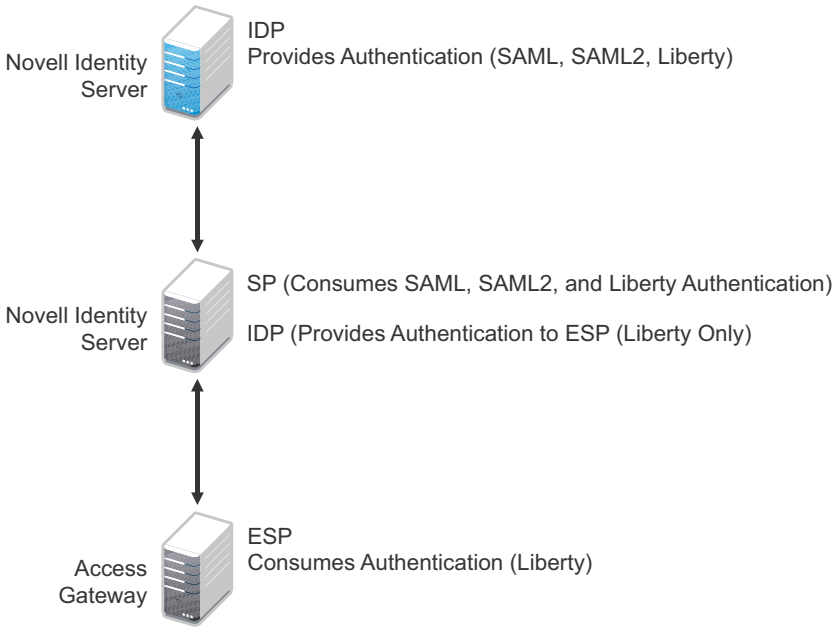
Setting up trust involves system administrators agreeing on how to establish a secure method for providing and consuming authentication assertions between their Identity Servers. An Identity Server is always installed as an identity provider, which is used to provide authentication to trusted service providers and ESPs. It can also be configured to be a service provider and trust the authentication of an identity provider.

- "Identity Providers and Consumers" on page 144
- "Embedded Service Providers" on page 145
- "Configuration Overview" on page 146

Identity Providers and Consumers

An Identity Server can be configured as an identity provider, which allows other service providers to trust it for authentication. It can also be configured as a service provider, which enables the Identity Server to consume authentication assertions from trusted identity providers. Figure 3-16 depicts two Identity Servers. The Identity Server at the top of the figure is configured as an identity provider for SAML 1.1, SAML 2.0, and Liberty authentication. The Identity Server in the middle of the figure is configured as a service provider, consuming the authentication credentials of the top Identity Server. This second Identity Server is also configured as an identity provider, providing authentication for the Embedded Service Provider of the Access Gateway.

Figure 3-16 *Identity Server Trust*

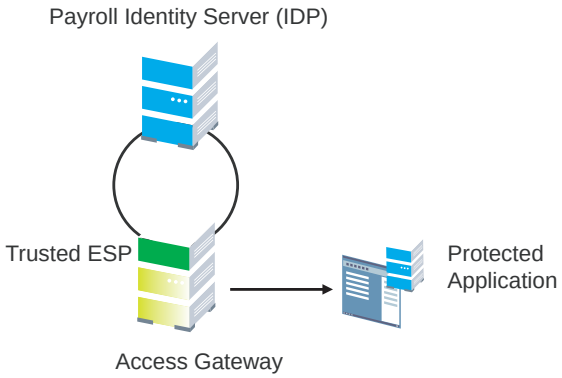


As an administrator, you determine whether your server is to be used as the identity provider or service provider in the trust relationship. You and the trusted partner agree to exchange identity provider metadata, and then you create references to the trusted partner's identity provider or service provider in your Identity Server configuration. You can obtain metadata via a URL or an XML document, then enter it in the system when you create the reference.

Embedded Service Providers

In addition to setting up trust with internal or external service providers, you can reference ESPs in your enterprise. An ESP uses the Liberty protocol and does not require metadata entry, because this exchange happens automatically. The ESP comes with Access Manager and is embedded in the Access Gateways ~~and a version of the SSL VPN server~~. The ESP facilitates authentication between the Identity Server and the resource protected by the device, as shown in as shown in Figure 3-17.

Figure 3-17 *Embedded Service Provider*



Configuration Overview

The following high-level tasks describe the process required to set up the trust model between an identity provider and a service provider. Although these tasks assume that both providers are Identity Servers provided with Access Manager, similar tasks must be performed when one of the providers is a third-party application.

1. Administrators at each company install and configure the Identity Server.
2. Administrators must exchange Identity Server metadata with the trusted partner.

   Metadata is generated by the Identity Server and can be obtained via a URL or an XML document, then entered in the system when you create the reference. This step is not applicable if you are referencing an ESP. When you reference an ESP, the system lists the installed ESPs for you to choose, and no metadata entry is required.
3. Create the reference to the trusted identity provider and the service provider.

   This procedure associates the metadata with the new provider. See “Creating a Trusted Service Provider” on page 152.
4. Configure user authentication.

   This procedure defines how your Identity Server interacts with the trusted provider during user authentication. Access Manager comes with default basic authentication settings already enabled. See Chapter , “Configuring Federated Authentication,” on page 372.

   Additional important steps for enabling authentication between trusted providers include:

   - Setting up the necessary authentication contracts. See “Configuring Authentication Contracts” on page 288.
   - Enabling the profiles that you are using. See “Managing Web Services and Profiles” on page 462.
   - Enabling the **Always Allow Interaction** option on the Web Service Consumer page. See “Configuring the Web Service Consumer” on page 471.
5. (Conditional) If you are setting up SAML 1.1 federation, the protocol does not allow the target link after federation to be automatically configured. You must manually configure this setting.

   See “Specifying the Intersite Transfer Service URL for the Login URL Option” on page 166.

Configuring General Provider Settings

The following settings are global. These affect any identity providers or identity consumers (service providers) that the Identity Server has been configured to trust:

- “Configuring the General Identity Provider Settings” on page 146
- “Configuring the General Identity Consumer Settings” on page 148
- “Configuring the Introductions Class” on page 148
- “Configuring the Trust Levels Class” on page 149

Configuring the General Identity Provider Settings

The following settings affect all identity providers that the Identity Server has been configured to trust.

1 In the Administration Console, click **Devices > Identity Servers > Edit > Identity Providers**.

2 To specify identity provider settings, fill in the following fields:

Show logged out providers: Displays logged-out providers on the identity provider's logout confirmation page.

Require Signed Authentication Requests: Specifies that for the Liberty 1.2 and SAML 2.0 protocols, authentication requests from service providers must be signed. When you enable this option for the identity provider, you must also enable the **Sign Authentication Requests** option under the **Identity Consumer** heading on this page for the external trusted service provider.

Use Introductions (Publish Authentications): Enables single sign-on from the service provider to the identity provider. The service provider determines the identity providers that users are already logged into, and then selectively and automatically asks for authentication from one of the identity providers. Introductions are enabled only between service and identity providers that have agreed to a circle of trust, which means that they have agreed upon a common domain name for this purpose.

After authenticating a user, the identity provider accesses a service at the service domain and writes a cookie to the common part of the service domain, publishing that the authentication has occurred.

Service Domain (Local and Common): Enables a service provider to access a service at the service domain prior to authenticating a user. This service reads cookies obtained at this domain and discovers if any identity providers have provided authentication to the user. The service provider determines whether any of these identity providers can authenticate a user without credentials. The service domain must resolve to the same IP address as the base URL domain.

For example, if an agreed-upon common domain is *xyz.com*, the service provider can specify a service domain of *sp.xyz.com*, and the identity provider can specify a service domain of *idp.xyz.com*. For the identity provider, *xyz.com* is the common value entered, and *idp* is the local value.

Port: The port to use for identity provider introductions. Port 8445 for HTTPS is the default and must be opened on your firewall. If you specify a different port, you must edit the Tomcat `server.xml` file.

3 Click **OK**, then update the Identity Server.

Configuring a Global White List of Target URLs

URL redirection, which many applications and services require, inherently brings in security risks. While redirecting, the request can be tampered to redirect users to an external, malicious site. To prevent such issues, you can configure a list of permissible domains. Redirection is allowed only to these configured domains.

1 Under Redirection White List, click **New**.

2 Specify **Domain**.

You can specify a domain name with an asterisk wildcard character (*) that represents the entire DNS subtree. For example, specifying `*.example.com` as a domain will allow redirection to all children domain under `examle.com` including `example.com`. The `WWW` prefix is not required. You can specify the `*` wildcard only at the lowest level of the subtree.

For example:

Valid domain name: *.example.com

Invalid domain name: `innerweb.*.com`.

You must configure at least one domain to prevent open redirection.

- Liberty: The `target` parameter is filtered. If the requested target is not the white list, the Identity Server does not login.

- WS-Fed: The `wreply` parameter is filtered. If the requested wreply is not in the white list, the Identity Server does not login. However, if `wreply` is same as the provider's single logout or single sign-on URL domain, the request is accepted.
- SAML2: For idpsend, the `target` parameter is filtered using this list. This list is not applicable for spsend.

3 Click **OK**.

Configuring the General Identity Consumer Settings

The following settings affect all identity consumers (service providers) that the Identity Server has been configured to trust.

1 In the Administration Console, click **Devices > Identity Servers** > **Edit** > **Identity Consumer**.

2 Specify whether the Identity Server can run as an identity consumer.

When the Identity Server is configured to run as an identity consumer, the Identity Server can receive (consume) authentication assertions from other identity providers.

Enable: Enables this site to function as service provider. This setting is enabled by default.

If this option is disabled, the Identity Server cannot trust or consume authentication assertions from other identity providers. You can create and enable identity providers for the various protocols, but they are not loaded or used until this option is enabled.

Require Signed Assertions: Specifies that all SAML assertions received by the service provider are signed by the issuing SAML authority. The signing authority uses a key pair to sign SAML data sent to this trusted provider.

Sign Authentication Requests: Specifies that the service provider signs authentication requests sent to an identity provider when using the Liberty 1.2 and SAML 2.0 protocols.

Use Introductions (Discover IDP Authentications): Enables a service provider to discover whether a user has authenticated to a trusted identity provider, so the user can use single sign-on without requiring authentication credentials.

- **Service domain:** The shared, common domain for all providers in the circle of trust. This domain must resolve to the same IP address as the base URL domain. You must enable the **Identity Consumer** option to enable this field.
- **Port:** The port to use for identity consumer introductions. Port 8446 for HTTPS is the default and must be opened on your firewall. If you specify a different port, you must edit the Tomcat `server.xml` file.

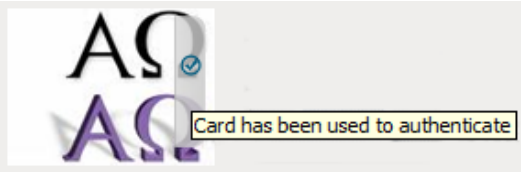
IMPORTANT: If you enable the **Use Introductions** option and you want to allow your users to select which identity provider to use for authentication rather than use single sign-on, you need to configure the Introductions class. See "Configuring the Introductions Class" on page 148.

---

3 Click **OK**, then update the Identity Server.

Configuring the Introductions Class

The Introduction class determines whether the user can select an identity provider to trust when the Identity Server is acting as a service provider. The default behavior is for introductions to happen automatically, thus allowing single sign-on. The Identity Server passively checks with the identity providers, one at time, to see if they can authenticate the service provider. If the identity provider can authenticate the user and the Introductions class is enabled, the user is presented with one or more cards that look similar to the following:



The small check mark indicates to the user that this is a possible card. When the user mouses over the card, the description appears. If the user selects one of these cards, the user is automatically authenticated.

To configure the Introductions class:

1. In the Administration Console, click **Devices > Identity Server > Servers > Edit > Local > Classes > Introductions**.
2. Click **Properties** > **New**, then specify the following values.

   Property Name: Specify `ShowUser`.

   Property Value: Specify `true`.
3. Click **OK**.
4. Return to the Servers page, then update the **Identity Server**.
5. When you configure this class, you need to also enable the **Use Introductions** option. Continue with "Configuring the General Identity Consumer Settings" on page 148.

Configuring the Trust Levels Class

The Trust Levels class allows you to specify an authentication level or rank for class types that do not appear on the Defaults page and for which you have not defined a contract. The level is used to rank the requested type. Using the authentication level and the comparison context, the Identity Server can determine whether any contracts meet the requirements of the request. If one or more contracts match the request, the user is presented with the appropriate authentication prompts. For more information and other configuration options, see "Specifying Authentication Defaults" on page 296 and "Specifying Authentication Types" on page 297

1. In the Administration Console, click **Devices > Identity Server > Servers > Edit > Local > Classes > Trust Levels**.
2. Click **Properties** > **New**, then specify the following values.

   Property Name: Specify `SetClassTrustLevels`.

   Property Value: Specify `true`.
3. For each class type for which you want to set a level, create a property for that class.

   3a Set the **Property Name** to the name of the class. For example, use one of the following:

   ```
   urn:oasis:names:tc:SAML:2.0:ac:classes:PreviousSession
   urn:oasis:names:tc:SAML:2.0:ac:classes:InternetProtocol
   ```

   For additional values, refer to the SAML2 and Liberty Authentication Context Specifications.

   3b Set the **Property Value** to the security level or rank you want for the class. A level of 2 is higher than a level of 1.
4. Click **OK**, then update the **Identity Server**.

Managing Trusted Providers

The procedure for establishing trust between providers begins with obtaining metadata for the trusted provider. If you are using the NetIQ Identity Server, protocol-specific metadata is available via a URL.

1 In the Administration Console, click **Devices > Identity Servers > Servers > Edit > [Protocol]**.

For the protocol, select Liberty, SAML 1.1 or SAML 2.0.

2 Select one of the following actions:

New: Launches the Create Trusted Identity Provider Wizard or the Create Trusted Service Provider Wizard, depending on your selection. See one of the following for more information:.

- “Creating a Trusted Service Provider” on page 152
- “Creating a Trusted Identity Provider” on page 150

Delete: Allows you to delete the selected identity or service provider.

Enable: Enables the selected identity or service provider.

Disable: Disables the selected identity or service provider. When a provider is disabled, the server does not load the definition. The definition is not deleted, and at a future time, the provider can be enabled.

IMPORTANT: When selecting which protocol to use, be aware of logout behavior of the SAML 1.1 protocol. The SAML 2.0 and Liberty 1.2 protocols define a logout mechanism whereby the service provider sends a logout command to the trusted identity provider when a user logs out at a service provider. SAML 1.1 does not provide such a mechanism. For this reason, when a logout occurs at the SAML 1.1 service provider, no logout occurs at the trusted identity provider. A valid session is still running at the identity provider, and no credentials need to be entered. In order to log out at both providers, the user must navigate to the identity provider that authenticated him to the SAML 1.1 service provider and log out manually.

NOTE: While adding the Access Manager Appliance as an identity provider or service provider to other Access Manager providers, the Metadata URL option must not be selected because Access Manager Appliance does not have any non-secure port for identity provider.

Creating a Trusted Identity Provider

Before you can create a trusted identity provider, you must complete the following tasks:

- Shared the trusted root of the SSL certificate of your Identity Server with the identity provider so that the administrator can imported it into the identity provider’s trust store.
- Obtained the metadata URL from the identity provider, an XML file with the metadata, or the information required for manual entry. For more information about the manual entry option, see “Editing a SAML 1.1 Identity Provider’s Metadata” on page 160.
- Shared the metadata URL of your Identity Server with the identity provider or an XML file with the metadata.
- Enabled the protocol. Click **Devices > Identity Servers > Edit**, and on the Configuration page, verify that the required protocol in the Enabled Protocols section has been enabled.

To create an identity provider:

1. In the Administration Console, click **Devices > Identity Servers > Servers > Edit > SAML 1.1**, **SAML 2.0**, or **Liberty**.
2. Click **New**, then click **Identity Provider**.
3. In the **Name** option, specify a name by which you want to refer to the provider.
4. Select one of the following sources for the metadata:

   Metadata URL: Specify the metadata URL for a trusted provider. The system retrieves protocol metadata using the specified URL. Examples of metadata URLs for an Identity Server acting as an identity provider with an IP address of 10.1.1.1:

   ```
   http://10.1.1.1:8080/nidp/saml/metadata
   https://10.1.1.1:8443/nidp/saml/metadata
   ```

   The nidp service is accelerated through the Access Gateway with the port 443. The nidp page can be accessed through /nidp directly without any port number. where nidp is the Tomcat application name.

   If your Identity Server and Administration Console are on different machines, use HTTP to import the metadata. If you are required to use HTTPS with this configuration, you must import the trusted root certificate of the provider into the trust store of the Administration Console. You need to use the Java `keytool` to import the certificate into the `cacerts` file in the security directory of the Administration Console.

   The `cacerts` file is located at:

   `/opt/novell/java/jre/lib/security`

   If you do not want to use HTTP and you do not want to import a certificate into the Administration Console, you can use the **Metadata Text** option. In a browser, enter the HTTP URL of the metadata. View the text from the source page, save the source metadata, then paste it into the **Metadata Text** option.

   Metadata Text: An editable field in which you can paste copied metadata text from an XML document, assuming you obtained the metadata via e-mail or disk and are not using a URL. If you copy metadata text from a Web browser, you must copy the text from the page source.

   Manual Entry: (SAML 1.1) Allows you to enter metadata values manually. When you select this option, the system displays the Enter Metadata Values page. See “Editing a SAML 1.1 Identity Provider’s Metadata” on page 160.

   Metadata Repositories: (SAML 1.1 and SAML 2.0) Allows you to configure several identity and/or service providers using a multi-entity metadata file available in a central repository.
5. Click **Next**.
6. Review the metadata certificates, then click **OK**.
7. Configure an authentication card to use with this identity provider. Fill in the following fields:

   ID: (Optional) Specify an alphanumeric value that identifies the card. If you need to reference this card outside of the Administration Console, you need to specify a value here. If you do not assign a value, the Identity Server creates one for its internal use

   Text: Specify the text that is displayed on the card to the user.

   Login URL: Specify an Intersite Transfer Service URL.The URL has the following format, where idp.sitea.novell.com is the DNS name of the identity provider and idp.siteb.novell.com is the name of the service provider:

   NOTE: The PID in the login URL must exactly match the entity ID specified in the metadata.

```
https://idp.sitea.novell.com:8443/nidp/saml/idpsend?PID=https://
idp.siteb.novell.com:8443/nidp/saml/metadata&TARGET=https://
idp.siteb.novell.com:8443/nidp/app
```

For more information, see “Specifying the Intersite Transfer Service URL for the Login URL Option” on page 166.

Image: Specify the image to be displayed on the card. Select the image from the drop down list. To add an image to the list, click **<Select local image>**.

Show Card: Determine whether the card is shown to the user, which allows the user to select and use the card for authentication. If this option is not selected, the card is only used when a service provider makes a request for the card.

8 Click **Finish**. The system displays the trusted provider on the protocol page.

9 Update the Identity Server.

The wizard allows you to configure the required options and relies upon the default settings for the other options. For information about how to configure the default settings and how to configure the other available options, see “Modifying a Trusted Provider” on page 154.

Creating a Trusted Service Provider

You can configure Identity Server to trust a service provider or an identity provider.

- When you create a trusted identity provider, you are allowing that identity provider to authenticate the user and Identity Server acts as a service provider.
- When you create a trusted service provider, you are configuring Identity Server to provide authentication for the service provider and Identity Server acts as an identity provider.

Both of these types of trust relationships require the identity provider to establish a trusted relationship with the service provider and the service provider to establish a trusted relationship with the identity provider.

The default settings of identity and service providers when you import the metadata repository are as follows:

- *SAML 1.1 Identity Provider*
  - Persistent Federation
  - Artifact Binding
  - No contracts associated to Satisfiable list of IDP
  - No image selected for the IDP card
  - Login URL will be empty with Show card disabled.
  - No attribute set associated
- *SAML 1.1 Service Provider*
  - No contracts associated to Satisfiable list of SP
  - No Attribute set associated
- *SAML 2.0 Identity Provider*
  - Persistent Federation as the Name Identifier
  - Artifact Binding
  - No contracts associated to Satisfiable list of IDP

- No image selected for the IDP card
- No Attribute set associated

- *SAML 2.0 Service Provider*
  - No contracts associated to Satisfiable list of SP
  - Artifact as Binding
  - No Attribute set associated

Prerequisites

Before you can create a trusted provider, you must complete the following tasks:

- Shared the trusted root of the SSL certificate of your Identity Server with the other provider so that the administrator can imported it into the provider's trust store.
- Obtained the metadata URL from the other provider or an XML file with the metadata.
- Shared the metadata URL of your Identity Server with the other provider or sent an XML file with the metadata.
- Enabled the protocol. Click **Devices > Identity Servers > Edit**, and on the Configuration page, verify that the required protocol in the Enabled Protocols section has been enabled.

Procedure

1. In the Administration Console, click **Devices > Identity Servers > Edit > [Protocol]**.
2. Click **New**, then click **Service Provider**.

   NOTE: By default, the **Provider Type > General** is selected. You can configure an Identity Server to trust a service provider to establish federation with external service providers. For more information on pre-configured metadata for Google Applications, Office 365, and Salesforce.com, see Chapter , "Configuring Authentication Through Federation for Specific Providers," on page 563.

   ---

3. Select one of the following sources for the metadata:

   Metadata URL: Specify the metadata URL for a trusted provider. The system retrieves protocol metadata by using the specified URL.

   Examples of metadata URLs for an Identity Server acting as a trusted provider with an IP address 10.1.1.1:

   - **Liberty:**

     `http://10.1.1.1:8080/nidp/idff/metadata`

     `https://10.1.1.1:8443/nidp/idff/metadata`
   - **SAML:**

     `http://10.1.1.1:8080/nidp/saml2/metadata`

     `https://10.1.1.1:8443/nidp/saml2/metadata`
   - **OIOSAML:**

     `http://10.1.1.1/nidp/saml2/metadata_oiosaml`

     `https://10.1.1.1/nidp/saml2/metadata_oiosaml`
   - **SAML 1.1:**

     `http://10.1.1.1:8080/nidp/saml/metadata`

     `https://10.1.1.1:8443/nidp/saml/metadata`

`/opt/novell/java/jre/lib/security`

Metadata Text: An editable field in which you can paste copied metadata text from an XML document, assuming you obtained the metadata via e-mail or disk and are not using a URL. If you copy metadata text from a Web browser, you must copy the text from the page source.

Manual Entry: Allows you to enter metadata values manually. When you select this option, the system displays the page to enter the required details. See “Editing a SAML 2.0 Service Provider’s Metadata” on page 159 or “Editing a SAML 1.1 Service Provider’s Metadata” on page 161.

Metadata Repositories: Allows you to configure several identity and/or service providers using a multi-entity metadata file available in a central repository.

4 In the **Name** option, specify a name by which you want to refer to the provider.

5 Click **Next**.

6 Review the metadata certificates and click **Finish**. The system displays the trusted provider on the protocol page.

7 Click **OK**, then update Identity Server.

The wizard allows you to configure the required options and relies upon the default settings for the other federation options. For information about how to configure the default settings and how to configure the other available options, see “Modifying a Trusted Provider” on page 154.

Modifying a Trusted Provider

The following sections describe the configuration options available for identity providers and service providers:

You can modify the following features of an identity provider:

- **Communication Security:** See “Communication Security” on page 155.
- **Attributes to Obtain at Authentication:** See “Configuring the Attributes Obtained at Authentication” on page 156.
- **Metadata:** See “Managing Metadata” on page 158.
- **Authentication Request:** See “Configuring a SAML 2.0 Authentication Request” on page 428 and “Configuring a Liberty Authentication Request” on page 458.
- **User Identification:** See Chapter , “Configuring Federated Authentication,” on page 372.
- **Authentication Card:** See “Modifying the Authentication Card for Liberty or SAML 2.0” on page 438 and “Modifying the Authentication Card for SAML 1.1” on page 454.

You can modify the following features of a service provider:

- **Communication Security:** See “Communication Security” on page 155.
- **Attributes to Send in the Response:** See “Configuring the Attributes Sent with Authentication” on page 156.
- **Intersite Transfer Service:** See “Configuring an Intersite Transfer Service Target for a Service Provider” on page 169.
- **Metadata:** See “Managing Metadata” on page 158.
- **Authentication Response:** See “Configuring an Authentication Response for a Service Provider” on page 162.

Communication Security

The communication security settings control the direct communication between the Identity Server and a trusted provider across the SOAP back channel. You can secure this channel with one of three methods:

Message Signing: This is the default method, and the Identity Server comes with a test signing certificate that is used to sign the back-channel messages. We recommend replacing this test signing certificate with a certificate from a well-known certificate authority. This method is secure, but it is CPU intensive. .

Mutual SSL: This method is probably the fastest method, and if you are fine-tuning your system for performance, you must select this method. However, it requires the exchange of trusted root certificates between the Identity Server and the trusted provider. This exchange of certificates is a requirement for setting up the trust relationship between the two providers. .

Basic Authentication: This method is as fast as mutual SSL and the least expensive because it does not' require any certificates. However, it does require the exchange of usernames and passwords with the administrator of the trusted provider, which might or might not compromise the security of the trusted relationship.

If your trusted provider is another Identity Server, you can use any of these methods, as long as your Identity Server and the trusted Identity Server use the same method. If you are setting up a trusted relationship with a third-party provider, you need to select a method supported by that provider.

For configuration information, see the following sections:

- "Configuring Communication Security for a SAML 2.0 Identity Provider" on page 426
- "Configuring Communication Security for a SAML 2.0 Service Provider" on page 427
- "Configuring Communication Security for SAML 1.1" on page 453
- "Configuring Communication Security for Liberty" on page 457

Selecting Attributes for a Trusted Provider

You can select attributes that an identity provider sends in an authentication request or that a service provider receives in an authentication response. The attributes are selected from an attribute set, which you can create or select from a list of already defined sets (see "Configuring Attribute Sets" on page 61).

For best performance, you must configure the trusted providers to use attribute sets, especially for attributes that have static values such as a user's e-mail address, employee ID, or phone number. It reduces the traffic between the provider and the LDAP server, because the attribute information can be gathered in one request at authentication rather than in a separate request for each attribute when a policy or protected resource needs the attribute information.

- "Configuring the Attributes Obtained at Authentication" on page 156
- "Configuring the Attributes Sent with Authentication" on page 156
- "Sending Attributes to the Embedded Service Provider" on page 157

Configuring the Attributes Obtained at Authentication

When the Identity Server creates its request to send to the identity provider, it uses the attributes that you have selected. The request asks the identity provider to provide values for these attributes. You can then use these attributes to create policies, to match user accounts, or if you allow provisioning, to create a user account on the service provider.

1. In the Administration Console, click **Devices > Identity Servers > Edit > [Protocol] > [Identity Provider] > Attributes**.
2. (Conditional) To create an attribute set, select **New Attribute Set** from the **Attribute Set** drop-down menu.

   An attribute set is a group of attributes that can be exchanged with the trusted provider. For example, you can specify that the local attribute of any attribute in the Liberty profile (such as Informal Name) matches the remote attribute specified at the service provider.

   2a Specify a set name, then click **Next**.

   2b On the Define Attributes page, click **New**.

   2c Select a local attribute.

   2d Optionally, provide the name of the remote attribute and a namespace.

   2e Click **OK**.

   For more information about this process, see “Configuring Attribute Sets” on page 61.

   2f To add other attributes to the set, repeat Step 2b through Step 2e.

   2g Click **Finish**.
3. Select an attribute set
4. Select attributes from the **Available** list, and move them to the left side of the page.

   The attributes that you move to the left side of the page are the attributes you want to be obtained during authentication.
5. Click **OK** twice.
6. Update the Identity Server.

Configuring the Attributes Sent with Authentication

When the Identity Server creates its response for the service provider, it uses the attributes listed on the Attributes page. The response needs to contain the attributes that the service provider requires. If you do not own the service provider, you need to contact the administrator of the service provider and negotiate which attributes you need to send in the response. The service provider can then use these attributes to identify the user, to create policies, to match user accounts, or if it allows provisioning, to create a user accounts on the service provider.

1. In the Administration Console, click **Devices > Identity Servers > Edit > [Protocol] > [Service Provider] > Attributes**.
2. (Conditional) To create an attribute set, select **New Attribute Set** from the **Attribute Set** drop-down menu.

   An attribute set is a group of attributes that can be exchanged with the trusted provider. For example, you can specify that the local attribute of any attribute in the Liberty profile (such as Informal Name) matches the remote attribute specified at the service provider.

   2a Specify a set name, then click **Next**.

   2b On the Define Attributes page, click **New**.

   2c Select a local attribute.

2d Optionally, you can provide the name of the remote attribute and a namespace.

2e Click **OK**.

For more information about this process, see "Configuring Attribute Sets" on page 61.

2f To add other attributes to the set, repeat Step 2b through Step 2e.

2g Click **Finish**.

3 Select an attribute set

4 Select attributes from the **Available** list, and move them to the left side of the page.

The left side of the page lists the attributes that you want sent in an assertion to the service provider.

5 Click **OK** twice.

6 Update the Identity Server.

Sending Attributes to the Embedded Service Provider

You can configure the Embedded Service Provider (ESP) of the Access Gateway to receive attributes when the user authenticates. LDAP traffic is reduced and performance is improved when the required LDAP attribute values are retrieved during authentication. This improvement is easily seen when you have many users and you have configured Identity Injection or Authorization policies to protect resources and these policies use LDAP attributes or Identity Server roles.

When the authentication process does not gather the LDAP attribute values, each user access can generate a new LDAP query, depending upon how the user accesses the resources and how the policies are defined. However, if the LDAP values are gathered at authentication, one LDAP query can retrieve all the needed values for the user.

1 In the Administration Console, click **Devices > Identity Servers** > Shared Settings.

2 On the Attributes page, click **New**, specify a name, then click **Next**.

3 For each attribute you need to add because it is used in a policy:

3a Click **New**.

3b In the **Local attribute** drop-down list, scroll to LDAP Attribute section, then select the attribute.

3c Click **OK**.

The other fields do not need to be configured.

4 If you use Identity Server roles in your policies, click **New**, select the All Roles attribute, then click **OK**.

5 Click **Finish**.

This saves the attribute set.

6 Click **Servers > Edit > Liberty**.

7 Click the name of the Embedded Service Provider.

If the Embedded Service Provider is part of a cluster of Access Gateways, the default name is the cluster name. If the Access Gateway is not part of a cluster, the default name is the IP address of the Access Gateway.

8 Click **Attributes**.

9 For the attribute set, select the set you created for the Embedded Service Provider.

10 Select attributes from the **Available** list, then move them to the left side of the page.

11 Click **OK**, then update the Identity Server.

Managing Metadata

The Liberty, SAML 1.1, and SAML 2.0 protocols contain pages for viewing and reimporting the metadata of the trusted providers.

- "Viewing and Reimporting a Trusted Provider's Metadata" on page 158
- "Viewing Trusted Provider Certificates" on page 159
- "Editing a SAML 2.0 Service Provider's Metadata" on page 159
- "Editing a SAML 1.1 Identity Provider's Metadata" on page 160
- "Editing a SAML 1.1 Service Provider's Metadata" on page 161

Viewing and Reimporting a Trusted Provider's Metadata

You might need to reimport a trusted provider's metadata if you learn that it has changed. The metadata changes when you change the provider to use HTTPS rather than HTTP and when you change the certificate that it is using for SSL. The steps for reimporting the metadata are similar for Liberty and SAML protocols.

NOTE: The trusted providers that are from the metadata repository cannot be reimported from this option. Go to **Shared Settings > > Metadata Repositories** and click on the metadata repository created to reimport the trusted provider.

---

1. In the Administration Console, click **Devices > Identity Servers > Edit > [Protocol]**.
2. Click the trusted provider, then click the **Metadata** tab.

   This page displays the current metadata the trusted provider is using.
3. To reimport the metadata:
   - **3a** Copy the URL in the providerID field (Liberty) or the entityID (SAML).
   - **3b** (SAML 1.1) Paste the URL to a file, click **Authentication Card**, copy the **Login URL** to the file, then click **Metadata**.
   - **3c** Click **Reimport**.
   - **3d** Follow the prompts to import the metadata.

     For the metadata URL, paste in the value you copied.

     If your Administration Console is installed with your Identity Server, you need to change the protocol from HTTPS to HTTP and the port from 8443 to 8080.
4. Confirm metadata certificates, then click **Finish**, or for an identity provider, click **Next**.
5. (Identity Provider) Configure the card, then click **Finish**.

   For SAML 1.1, copy the value you saved into the **Login URL**.
6. Update the Identity Server.

NOTE: Reimport support is not available for SAML 1.1 and SAML 2.0 protocols.

Viewing Trusted Provider Certificates

You can review and confirm the certificate information for identity and service providers.

1. In the Administration Console, click **Devices** > **Identity Servers > Edit > [Protocol] > [Name of Provider] > Metadata > Certificates**.
2. View the following information is displayed for the certificates:

   Subject: The subject name assigned to the certificate.

   Validity: The first date the certificate was valid, and the date the certificate expires.

   Issuer DN: The distinguished name of the Certificate Authority (CA) that created the certificate.

   Algorithm: The name of the algorithm that was used to create the certificate.

   Serial Number: The serial number that the CA assigned to the certificate.
3. Click **OK** if you are viewing the information, or click **Next** or **Finish** if you are creating a provider.

Editing a SAML 2.0 Service Provider's Metadata

Access Manager allows you to obtain metadata for SAML 2.0 providers. However, metadata for SAML 2.0 might not be available for some service providers, so you can enter the metadata manually. The page for this is available if you clicked the **Manual Entry** option when you created the trusted provider.

1. In the Administration Console, click **Devices > Identity Servers > Edit > SAML 2.0 > [Service Provider] > Metadata**.

   You can reimport the metadata (see Step 2) or edit it (see Step 3).
2. To reimport the metadata, click **Reimport** on the View page.

   Follow the on-screen instructions to complete the steps in the wizard.
3. To edit the metadata manually, click **Edit**.
4. Fill in the following fields:

   Provider ID: (Required) Specifies the SAML 2.0 metadata unique identifier for the provider. For example, `https://<dns>:8443/nidp/saml2/metadata`. Replace <*dns*> with the DNS name of the provider.

   In the metadata, this is the entityID value.

   Metadata expiration: Specifies the date upon which the metadata is no longer valid.

   Want assertion to be signed: Specifies that authentication assertions from the trusted provider must be signed.

   Artifact consumer URL: Specifies where the partner receives incoming SAML artifacts. For example, `https://<dns>:8443/nidp/saml2/spassertion_consumer`. Replace <*dns*> with the DNS name of the provider.

   In the metadata, this URL value is found in the AssertionConsumerService section of the metadata.

   Post consumer URL: Specifies where the partner receives incoming SAML POST data. For example, `https://<dns>:8443/nidp/saml2/spassertion_consumer`. Replace <*dns*> with the DNS name of the provider.

   In the metadata, this URL value is found in the AssertionConsumerService section of the metadata.

Service Provider: Specifies the public key certificate used to sign SAML data. You can browse to locate the service provider certificate.

5 Click **Finish**.

Editing a SAML 1.1 Identity Provider's Metadata

Access Manager allows you to import metadata for SAML 1.1 providers. However, metadata for SAML 1.1 might not be available for some trusted providers, so you can enter metadata manually. The page for this is available if you clicked the **Manual Entry** option when you created the trusted provider.

1 In the Administration Console, click **Devices > Identity Servers > Edit > SAML 1.1 > [Identity Provider] > Metadata**.

You can reimport the metadata (see Step 2) or edit it (see Step 4).

2 To reimport the metadata from a URL or text, click **Reimport** on the View page.

The system displays the Create Trusted Identity Provider Wizard that lets you obtain the metadata. Follow the on-screen instructions to complete the steps in the wizard.

3 Select either **Metadata URL** or **Metadata Text**, then fill in the field for the metadata.

4 To edit the metadata manually, click **Edit**.

5 Fill in the following fields as necessary:

Supported Version: Specifies the version of SAML that you want to use. You can select SAML 1.0, SAML 1.1, or both SAML 1.0 and SAML 1.1.

Provider ID: (Required) The SAML 1.1 metadata unique identifier for the provider. For example, `https://<dns>/nidp/saml/metadata`. Replace <*dns*> with the DNS name of the provider.

In the metadata, this is the entityID value.

Source ID: The SAML Source ID for the trusted provider. The Source ID is a 20-byte value that is used as part of the Browser/Artifact profile. It allows the receiving site to determine the source of received SAML artifacts. If none is specified, the Source ID is auto-generated by using a SHA-1 hash of the site provider ID.

Metadata expiration: The date upon which the metadata is no longer valid.

SAML attribute query URL: The URL location where an attribute query is to be sent to the partner. The attribute query requests a set of attributes associated with a specific object. A successful response contains assertions that contain attribute statements about the subject. A SAML 1.1 provider might use the base URL, followed by /saml/soap. For example, `https://<dns>:8443/nidp/saml/soap`. Replace <*dns*> with the DNS name of the provider.

In the metadata, this URL value is found in the AttributeService section of the metadata.

Artifact resolution URL: The URL location where artifact resolution queries are sent. A SAML artifact is included in the URL query string. The target URL on the destination site the user wants to access is also included on the query string. A SAML 1.1 provider might use the base URL, followed by /saml/soap. For example, `https://<dns>:8443/nidp/saml/soap`. Replace <*dns*> with the DNS name of the provider.

In the metadata, this URL value is found in the ArtifactResolutionService section of the metadata.

6 To specify signing certificate settings, fill in the followi

ng fields:

Attribute authority: Specifies the signing certificate of the partner SAML 1.1 attribute authority. The attribute authority relies on the identity provider to provide it with authentication information so that it can retrieve attributes for the appropriate entity or user. The attribute authority must know that the entity requesting the attribute has been authenticated to the system.

Identity provider: (Required) Appears if you are editing identity provider metadata. This field specifies the signing certificate of the partner SAML 1.1 identity provider. It is the certificate the partner uses to sign authentication assertions.

7 Click **OK**.

8 On the Identity Servers page, click **Update All** to update the configuration.

Editing a SAML 1.1 Service Provider's Metadata

Access Manager allows you to obtain metadata for SAML 1.1 providers. However, metadata for SAML 1.1 might not be available for some trusted providers, so you can enter the metadata manually. The page for this is available if you clicked the **Manual Entry** option when you created the trusted provider.

For conceptual information about how Access Manager uses SAML, see Chapter , "Understanding How Access Manager Uses SAML," on page 418.

1 In the Administration Console, click **Devices > Identity Servers > Edit > SAML 1.1 > [Service Provider] > Metadata**.

You can reimport the metadata (see Step 2) or edit it (see Step 3).

2 To reimport the metadata, click **Reimport** on the View page.

Follow the on-screen instructions to complete the steps in the wizard.

3 To edit the metadata manually, click **Edit**.

4 Fill in the following fields:

Supported Version: Specifies which version of SAML that you want to use. You can select SAML 1.0, SAML 1.1, or both SAML 1.0 and SAML 1.1.

Provider ID: (Required) Specifies the SAML 1.1 metadata unique identifier for the provider. For example, `https://<dns>:8443/nidp/saml/metadata`. Replace <*dns*> with the DNS name of the provider.

In the metadata, this is the entityID value.

Metadata expiration: Specifies the date upon which the metadata is no longer valid.

Want assertion to be signed: Specifies that authentication assertions from the trusted provider must be signed.

Artifact consumer URL: Specifies where the partner receives incoming SAML artifacts. For example, `https://<dns>:8443/nidp/saml/spassertion_consumer`. Replace <*dns*> with the DNS name of the provider.

In the metadata, this URL value is found in the AssertionConsumerService section of the metadata.

Post consumer URL: Specifies where the partner receives incoming SAML POST data. For example, `https://<dns>:8443/nidp/saml/spassertion_consumer`. Replace <*dns*> with the DNS name of the provider.

In the metadata, this URL value is found in the AssertionConsumerService section of the metadata.

Service Provider: Specifies the public key certificate used to sign SAML data. You can browse to locate the service provider certificate.

5 Click **Finish**.

Configuring an Authentication Response for a Service Provider

The Liberty and SAML 2.0 protocols support slightly different options for configuring how you want the Identity Server to respond to an authentication request from a service provider. The SAML 1.1 protocol does not support sending an authentication request. However, you can configure an Intersite Transfer Service (see "Using the Intersite Transfer Service" on page 164) to trigger a response from the Identity Server.

When the Identity Server receives an authentication request from a trusted service provider, the request contains the conditions that the Identity Server needs to fulfill. The Authentication Response page allows you to configure how you want the Identity Server to fulfill the binding and name identifier conditions of the request, or for SAML 1.1, respond to the Intersite Transfer Service. For configuration information, see one of the following:

- "Configuring the Liberty Authentication Response" on page 460
- "Configuring the SAML 2.0 Authentication Response" on page 431
- "Configuring the SAML 1.1 Authentication Response" on page 453

The Defaults page allows you to specify which contract is used when the authentication request specifies a class or type rather than a contract. For more information, see "Specifying Authentication Defaults" on page 296.

When the service provider sends an authentication request that specifies a specific contract, you need to ensure that the Identity Server has a the contract matches the expected URI. For information about how to configure such a contract, see "Creating a Contract for a Specific Authentication Type" on page 297.

Routing to an External Identity Provider Automatically

When the NetIQ Identity Server is configured to federate with multiple external Identity Providers, administrator can specify the list of Authentication Contracts that an external provider can execute. This configuration allows the NetIQ Identity Server (acting as service provider) to automatically select the external identity provider without the user having to click on the external provider's card.

Authentication Contracts in the NetIQ Identity Servers have been enhanced to be configured with an Authentication Class Reference. This reference can be used in federating with External Identity or Service Providers that only respond to `AuthnContextClassRef` in the Authentication Request and Response. For more information about setting up the contract mapping and adding contracts to the satisfiable list, see "Modifying the Authentication Card for Liberty or SAML 2.0" on page 438 and "Configuring Authentication Contracts" on page 288.

Configuring Options for Trusted Service Providers

After you create a trusted service provider, you can configure how your Identity Server responds to authentication requests from the service provider.

1. In the Administration Console, click **Devices > Identity Servers > Edit > SAML 2.0 > [Service Provider] > Authentication Response**.
2. Select the binding method.

   If the request from the service provider does not specify a response binding, you need to specify a binding method to use in the response. Select **Artifact** to provide an increased level of security by using a back-channel means of communication between the two servers. Select **Post** to use HTTP redirection for the communication channel between the two servers. If you select **Post**, you might want to require the signing of the authentication requests. See "Configuring the General Identity Provider Settings" on page 146.
3. Specify the identity formats that the Identity Server can send in its response. Select the box to choose one or more of the following:
   - **Persistent:** Specifies that a persistent identifier, which is written to the directory and remains intact between sessions, can be sent.
   - **Transient:** Specifies that a transient identifier, which expires between sessions, can be sent.
   - **E-mail:** Specifies that an e-mail attribute can be used as the identifier.
   - **Kerberos:** Specifies that a Kerberos token can be used as the identifier.
   - **X509:** Specifies that an X.509 certificate can be used as the identifier.
   - **Unspecified:** Specifies that an unspecified format can be used and any value can be used. The service provider and the identity provider need to agree on the value that is placed in this identifier.
4. Use the **Default** button to select the name identifier that the Identity Server must send if the service provider does not specify a format.

   If you select E-mail, Kerberos, x509, or unspecified as the default format, you must also select a value. See Step 5.

   IMPORTANT: If you have configured the identity provider to allow a user matching expression to fail and still allow authentication by selecting the **Do nothing** option, you need to select **Transient identifier format** as the default value. Otherwise the users who fail the matching expression are denied access. To view the identity provider configuration, see "Defining User Identification for Liberty and SAML 2.0" on page 411.

   ---
5. Specify the value for the name identifier.

   The persistent and transient formats are generated automatically. For the others, you can select an attribute. The available attributes depend upon the attributes that you have selected to send with authentication (see "Configuring the Attributes Obtained at Authentication" on page 156). If you do not select a value for the E-mail, Kerberos, X509, or Unspecified format, a unique value is automatically generated.
6. To specify that this Identity Server must authenticate the user, disable the **Use proxied requests** option. When the option is disabled and the Identity Server cannot authenticate the user, the user is denied access.

When this option is enabled, the Identity Server checks to see if other identity providers can satisfy the request. If one or more can, the user is allowed to select which identity provider performs the authentication. If a proxied identity provider performs the authentication, it sends the response to the Identity Server. The Identity Server then sends the response to the service provider.

7 Click **OK** twice, then update the Identity Server.

Using the Intersite Transfer Service

- "Understanding the Intersite Transfer Service URL" on page 164
- "Specifying the Intersite Transfer Service URL for the Login URL Option" on page 166
- "Using Intersite Transfer Service Links on Web Pages" on page 167
- "Configuring an Intersite Transfer Service Target for a Service Provider" on page 169
- "Configuring Whitelist of Target URLs" on page 170
- "Validating Incoming Authentication Request for Assertion Consumer Service URL" on page 170
- "Federation Entries Management" on page 171
- "Step up Authentication Example for an Identity Provider Initiated Single Sign-On Request" on page 171
- "URL Query String Parameters" on page 173

Understanding the Intersite Transfer Service URL

The Intersite Transfer Service is used by an identity provider to provide authentication to occur at a service provider that it trusts. The URLs for accessing the Intersite Transfer Service differ for each supported protocol (Liberty, SAML 1.1, and SAML 2.0). The NetIQ Access Manager identity and service provider components use the following format of the Intersite Transfer Service URL:

`<identity provider URL>?PID=<entityID>&TARGET=<final_destination_URL>`

The *<identity provider URL>* is the location where the authentication request can be processed. For an Access Manager Identity Server, the URL is the Base URL of the server that provides authentication, followed by the path to the protocol application being used for federation.

For example:

SAML 1.1: `https://idp.sitea.novell.com:8443/nidp/saml/idpsend`

SAML 2.0: `https://idp.sitea.novell.com:8443/nidp/saml2/idpsend`

Liberty: `https://idp.sitea.novell.com:8443/nidp/idff/idpsend`

If a third-party server provides the authentication, refer the documentation for the format of this URL.

The *<entityID>* is the URL to the location of the metadata of the service provider. The scheme (http or https) in the *<entityID>* must match what is configured for the *<identity provider URL>*.

For SAML 1.1 and SAML 2.0, search the metadata for its entityID value. For Liberty, search the metadata for the providerID value. Access Manager Identity Servers acting as service providers have the following types of values:

SAML 1.1: `https://idp.siteb.novell.com:8443/nidp/saml/metadata`

SAML 2.0: `https://idp.siteb.novell.com:8443/nidp/saml2/metadata`

Liberty: `https://idp.siteb.novell.com:8443/nidp/idff/metadata`

If you are setting up federations with a third-party service provider, refer the documentation for the URL or location of its metadata.

The <*final_destination_URL*> is the URL to which the browser is redirected following a successful authentication at the identity provider. If this target URL contains parameters (for example, `TARGET=https://login.provo.novell.com:8443/nidp/app?function_id=22166&Resp_Id=55321 &Resp_App_Id=810&security_id=0`), the URL must be encoded to prevent it from being truncated.

For example:

- **SAML 1.1:** `https://idp.sitea.novell.com:8443/nidp/saml/idpsend?PID=https://idp.siteb.novell.com:8443/nidp/saml/metadata&TARGET=https://eng.provo.novell.com/saml1/myapp`
- **SAML 2.0:** `https://idp.sitea.novell.com:8443/nidp/saml2/idpsend?PID=https://idp.siteb.novell.com:8443/nidp/saml2/metadata&TARGET=https://eng.provo.novell.com/saml2/myapp`
- **Liberty:** `https://idp.sitea.novell.com:8443/nidp/idff/idpsend?PID=https://idp.siteb.novell.com:8443/nidp/idff/metadata&TARGET=https://eng.provo.novell.com/liberty/myapp`

To read more about configuring an intersite URL, see "Configuring an Intersite Transfer Service Target for a Service Provider" on page 169.

If you configure an Intersite Transfer Service URL for an Identity Server that is the Access Manager Identity Server and the service provider is either another Identity Server or a third-party server, you can simplify the Intersite Transfer Service URL to the following format:

`<identity provider URL>?id=<user_definedID>`

For example:

- **SAML 2.0:** `https://test.blr.novell.com:8443/nidp/saml2/idpsend?id=testsaml2&TARGET=https://eng.provo.novell.com`
- **SAML 1.1:** `https://testsb.blr.novell.com:8443/nidp/saml/idpsend?id=testsaml&TARGET=https://eng.provo.novell.com`
- **Liberty:** `https://testsb.blr.novell.com:8443/nidp/idff/idpsend?id=libertytest&TARGET=https://eng.provo.novell.com`

If the **Allow any target** option is enabled and if the Intersite Transfer Service URL has a target value, then the user is redirected to target URL.

The Intersite Transfer Service URL for SAML 2.0 will be `https://testsb.blr.novell.com:8443/nidp/saml2/idpsend?id=testsaml2&TARGET=http://www.google.com` where `http://www.google.com` is the target URL.

NOTE: Depending on the usage, the target parameter serves different purpose. It is case-sensitive.

- **target:** Specifies the idpsend URL with a contract id.
- **TARGET:** Specifies URL of the final destination.

Use case: If authentication with a particular contract is enabled in Intersite URL, you are redirected to the default target URL. Use the following format: `http(s)://<$idp_host_name>/nidp/app?id=<$contract_to_be_executed>&target=http(s)://<$idp_host_name>/nidp/saml2/idpsend?id=<$saml_sp_identifier>`.

For more information, see How to access an Identity Server Intersite Transfer URL with a specific contract.

NOTE: The `contract_to_be_executed` is executed by the Identity Server and is case sensitive.

For example, `https://www.idp.com:8443/nidp/app?id=npbasic&target=https://www.idp.com:8443/nidp/saml2/idpsend?id=serviceprovider1`.

How it works?

In the above example, identity provider authentication is done with the contract id `npbasic`. You are now redirected to the service provider by using the `saml_sp_identifier id` (serviceprovider1). After authentication (if configured with persistent federation), the page redirects you to the available default target, or to the nidp login page of the service provider.

For configuration and ID information, see "Configuring an Intersite Transfer Service Target for a Service Provider" on page 169.

In the Intersite Transfer Service URL, id can be used for the following purposes:

1. To simplify the intersite URL.<identity provider URL>?id=<user_definedID>
2. To execute a particular contract in the Identity Server login service with intersite URL.

   ```
   http(s)://<$idp_host_name>/nidp/
   app?id=<$contract_to_be_executed>&target=http(s)://<$idp_host_name>/nidp/
   saml2/idpsend?id=<$saml_sp_identifier>
   ```

Specifying the Intersite Transfer Service URL for the Login URL Option

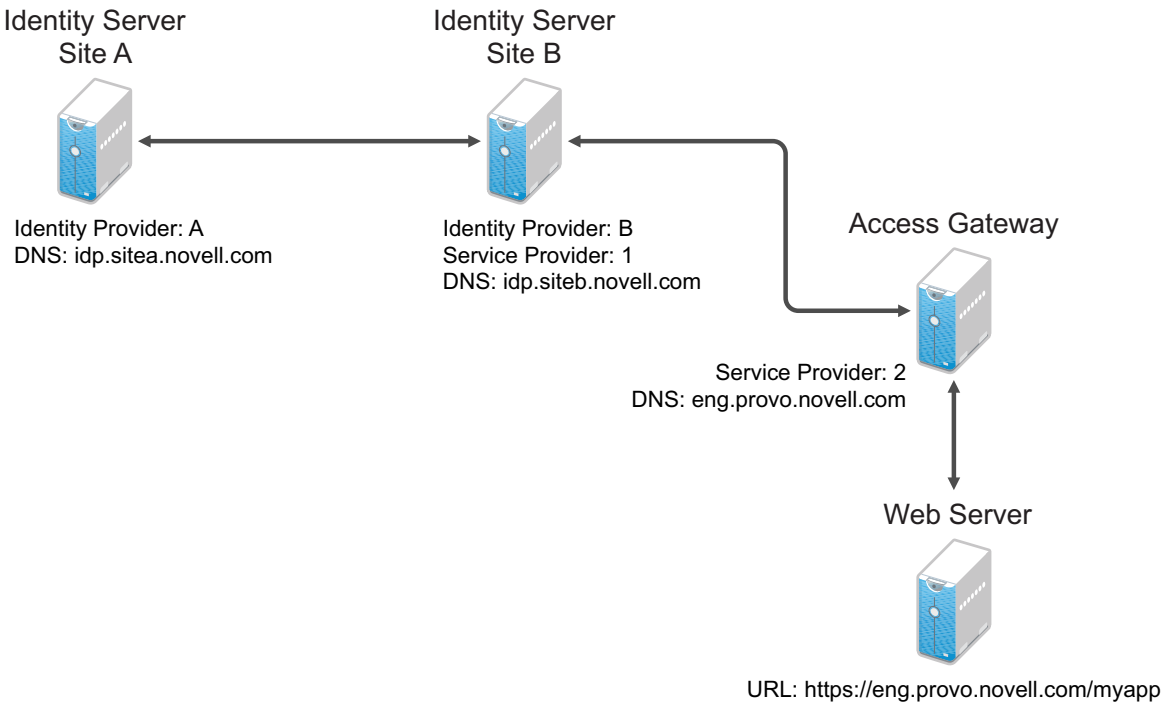
Liberty and SAML 2.0 support a single sign-on URL. Because SAML 1.1 does not support a single sign-on URL, you need to specify the Intersite Transfer Service URL in the **Login URL** option on the authentication card for the SAML 1.1 identity provider:

Figure 3-18 *SAML 1.1 Authentication Card*



For a card to appear as a login option, you must specify a **Login URL** and select the **Show Card** option. Figure 3-19 illustrates a possible configuration that requires the Intersite Transfer Service for the SAML 1.1 protocol.

Figure 3-19 *Federated Identity Configuration*



If you want a card to appear that allows the user to log in to Site A (as shown in Figure 3-18), you need to specify a value for the **Login URL** option.

Using the DNS names from Figure 3-19, the complete value for the **Login URL** option is as follows:

```
https://idp.sitea.novell.com:8443/nidp/saml/idpsend?PID=https://
idp.siteb.novell.com:8443/nidp/saml/metadata&TARGET=https://
idp.siteb.novell.com:8443/nidp/app
```

The following happens when this link is invoked:

1. The browser performs a Get to the identity provider (Site A).
2. If the identity provider (Site A) trusts the service provider (Site B), the identity provider prompts the user for authentication information and builds an assertion.
3. The identity provider (Site A) sends the user to the service provider (Site B), using the POST or Artifact method.
4. The service provider (Site B) consumes the assertion and sends the user to the TARGET URL (the user portal on Site B).

To configure the settings for the intersite transfer service, see “Modifying the Authentication Card for SAML 1.1” on page 454.

Using Intersite Transfer Service Links on Web Pages

The Intersite Transfer Service URL can be used on a Web page that provides links to various protected resources requiring authentication with a specific identity provider and a specific protocol. Links on this Web page are configured with the URL of the Intersite Transfer Service of the identity provider to be used for authentication. Clicking these links directs the user to the appropriate identity provider for authentication. Following successful authentication, the identity provider sends a SAML

assertion to the service provider. The service provider uses the SAML assertion to verify authentication, and then redirects the user to the destination URL as specified in the TARGET portion of the Intersite Transfer Service URL.

Below are sample links that might be included on a Web page. These links demonstrate the use of SAML 1.1, SAML 2.0, and Liberty formats for the Intersite Transfer Service URL:

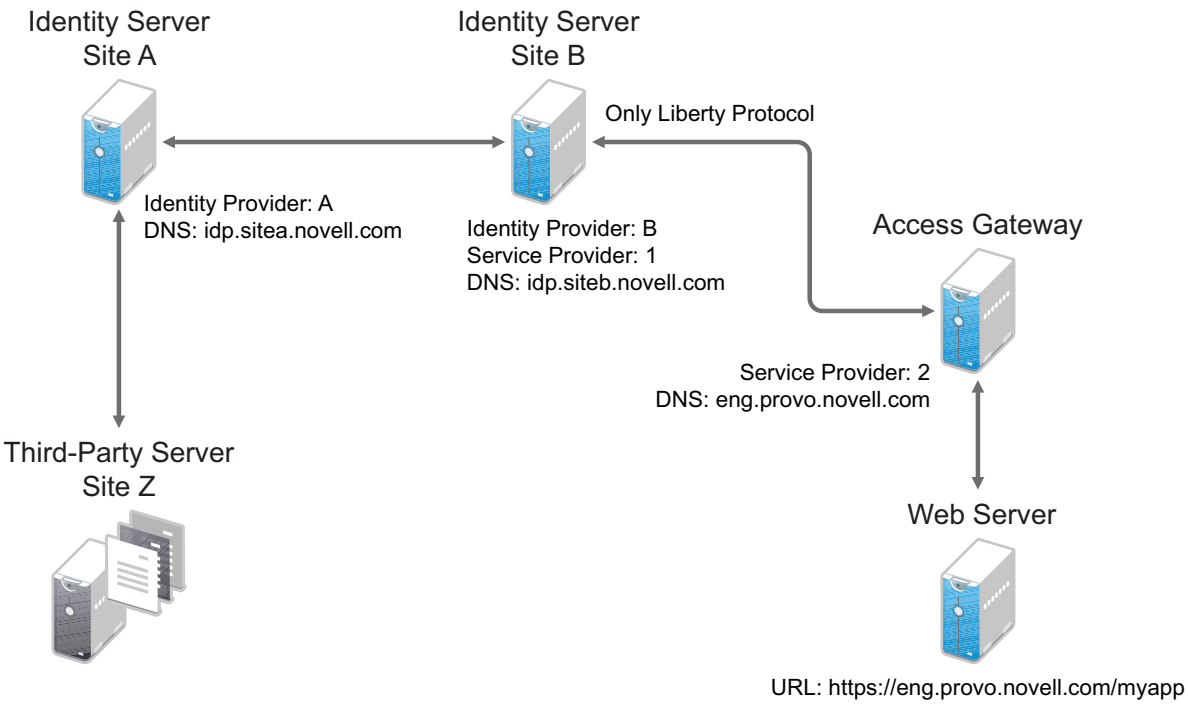
SAML 1.1: `<a href="https://idp.sitea.novell.com:8443/nidp/saml/idpsend?PID=https://idp.siteb.novell.com:8443/nidp/saml/metadata&TARGET=https://eng.provo.novell.com/saml1/myapp">SAML1 example</a>`

SAML 2.0: `<a href="https://idp.sitea.novell.com:8443/nidp/saml2/idpsend?PID=https://idp.siteb.novell.com:8443/nidp/saml2/metadata&TARGET=https://eng.provo.novell.com/saml2/myapp">SAML2 example</a>`

Liberty: `<a href="https://idp.sitea.cit.novell.com:8443/nidp/idff/idpsend?PID=https://idp.siteb.novell.com:8443/nidp/idff/metadata&TARGET=https://eng.provo.novell.com/liberty/myapp">Liberty example</a>`

Figure 3-20 illustrates a network configuration that could use these sample links.

Figure 3-20 *Using the Intersite Transfer Service URL*



In this example, Site Z places links on its Web page, using the Intersite Transfer Service URL of Site A. These links trigger authentication at Site A. If authentication is successful, Site A sends an assertion to Site B. Site B verifies the authentication and redirects the user to the myapp application that it is protecting.

When defining the intersite transfer URL within the Administration Console, you can define an id and target for the SAML service provider (SP) you are accessing. For more information about accessing an Identity Server intersite transfer URL with a specific contract, see TID 7005810.

Configuring an Intersite Transfer Service Target for a Service Provider

If you have created Web pages that have links that specify a Intersite Transfer Service URL (see "Using Intersite Transfer Service Links on Web Pages" on page 167), you can have the Identity Server control the TARGET parameter.

1. Click **Devices** > **Identity Servers** > **Edit** > **[Liberty, SAML1.1**, or **SAML 2.0]** > **[Service Provider]** > **Intersite Transfer Service**.
2. Fill in the following:

   ID: (Optional) Specify an alphanumeric value that identifies the target.

   If you specified an ID for the target, you can use this value to simplify the Intersite Transfer URL that must be configured at the service provider. This is the <*user_definedID*> value in the following format for the Intersite Transfer URL.

   `<identity provider URL>?id=<user_definedID>`

   The ID specified here allows the Identity Server to find the service provider's metadata.

   Target: Specify the URL of the page that you want to display to users when they authenticate with an Intersite Transfer URL.The behavior of this option is influenced by the **Allow any target** option. NetIQ recommends you to specify a default target URL, for example, https://www.serviceprovider1.com.

   Allow any target: You can either select or not select this option.

   - When you select this option,
     - if the Intersite Transfer URL has a target value, then the user is sent to target url.
     - if the Intersite Transfer URL does not have a target value, then the user is sent to the configured target, that is, www.serviceprovider1.com.
   - When you do not select this option,
     - if the Intersite Transfer URL has a target value, then the user is sent to the target www.serviceprovider1.com irrespective of the target mentioned in the Intersite Transfer URL.
     - if the Intersite Transfer URL does not have a target value, the user is sent to www.serviceprovider1.com.
3. Click **OK** twice.
4. Update the Identity Server.

Configuring Whitelist of Target URLs

Redirection, which is required by many applications and services, inherently brings in a security risk. Redirects are dangerous because unsuspecting users who are visiting trusted sites can be redirected to malicious sites that exploit the users' trust. A new featured, called whitelist, has been added that restricts target URLs to specific domains.

The whitelist feature allows you to restrict target URLs to URLs which match the domains in the whitelist. Any target URLs that use a domain that is not in the list are blocked and the user receives the following error message: `The request to provide authentication to a service provider has failed (outsidedomain.com-89F57BF823DFE551)`.

1. Click **Devices** > **Identity Servers** > **Edit** > **[Liberty, SAML1.1**, or **SAML 2.0]** > **[Service Provider]** > **Intersite Transfer Service**.
2. In the **Domain List**, click **New**.
3. Enter the domain name, then click **OK**.

   The domain name must be a full domain name, such as `www.novell.com`. Wildcard domain names, such as `www.novell.*.com`, do not work.
4. To edit an existing domain name, click the name, modify the name, then click **OK**.
5. To delete an existing domain name, select the check box by the domain, click **Delete**, then click **OK** to delete or **Cancel** to close the window.
6. Click **OK**.
7. Update the Identity Server.

Validating Incoming Authentication Request for Assertion Consumer Service URL

When an authentication request from a service provider is not signed, Identity Provider cannot validate the authenticity and integrity of the request. So any malicious user who can intercept the request can change the Assertion Consumer Service URL in the request and make the Identity Provider to send the assertion to malicious sites.

To secure and validate the authentication request from the service provider, you can use the following options in the service provider configuration of Identity provider:

NOTE: These options must be defined to avoid security issues during an unsigned SAML Authentication Request.

SAML2_ACS_URL_RESTRICT : This option ensures Identity Provider will validate the Assertion Consumer Service URL in the request against the trusted metadata URL before sending the assertion. So if the Assertion Consumer URL in the Authentication Request is tampered by any malicious user, Identity Provider terminates the request and assertion will not be sent.

SAML2_ACS_DOMAIN_WHITELIST: This option ensures Identity Provider will validate the Assertion Consumer Service URL in the request against a white list of domains. If the Assertion Consumer Service URL is not matching with any of the domain URLs in the white list, request is terminated by the Identity Provider.

You must define **SAML2_ACS_DOMAIN_WHITELIST** along with **SAML_ACS_URL_RESTRICT** for a service provider in Identity Provider because this option does not work if **SAML_ACS_URL_RESTRICT** is not enabled.

To define these options, perform the following in Administration Console:

1. Click **Devices > Identity Servers > IdP Cluster > SAML2**.
2. Select the required service provider from the **Service Providers** list.
3. Click **Options**.
4. Click **New**, then select **OTHER** from the drop down list.
   - **4a** If you want Identity Provider to allow authentication only to the trusted ACS URLs, specify the following:

     Property Name: **SAML2_ACS_URL_RESTRICT**

     Property Value: **true**
   - **4b** If you want Identity Provider to perform additional validation of the authentication request with the ACS domain whitelist, specify the following:

     Property Name: **SAML2_ACS_DOMAIN_WHITELIST**

     Property Value: Domain names separated with semi-colon(;) and no space. For example, *www.airlines.com*;*www.example.com*.

Federation Entries Management

Identity federation is the association of accounts between an identity provider and a service provider.

Step up Authentication Example for an Identity Provider Initiated Single Sign-On Request

Setup: Let us assume that:

- NetIQ Access Manager is acting as an identity provider.
- The following three contracts in the identity provider are configured:
  - name password basic contract with Authentication level as 10
  - name password form contract with Authentication level as 20
  - secure name password contract with Authentication level as 30

  NOTE: Enable the Satisfiable by a contract of equal or higher level option for contracts with authentication level 10 or 20 to avoid prompting for authentication when a user is already authenticated against the contract with level 30.

- The name password form contract for a service provider named SP_A is configured in the identity provider.

  For more information about creating and configuring the contracts, see “Configuring Authentication Contracts” on page 288.

Configuration: Complete the following steps:

1. In the NetIQ Identity Server, configure the service provider as a trusted provider.

   For more information, see “Managing Trusted Providers” on page 150.
2. In the service provider, configure the NetIQ Identity Server as a trusted provider.

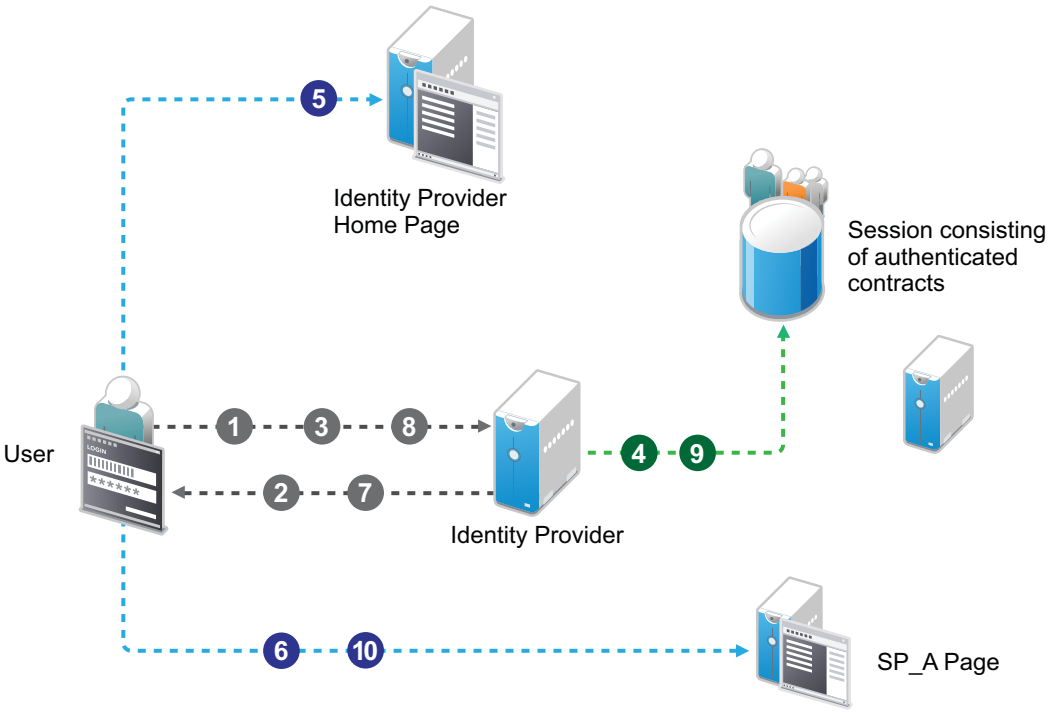
   For more information, see “Managing Trusted Providers” on page 150.
3. In the NetIQ Identity Server, configure the service provider with the required authentication contracts.

For information about how to configure a service provider, see “Defining Options for SAML 2.0” on page 433, “To Define Options for Liberty Service Provider” on page 461 and“Defining Options for SAML 1.1 Service Provider” on page 454.

Results: The following are the four possible scenarios:

- If the user was authenticated with the name password basic contract before making an Intersite Transfer Service request to SP_A, the identity provider will step up to the name password form authentication.
- If the user was authenticated with the name password form contract before making an Intersite Transfer Service request to SP_A, the identity provider will not ask for the authentication.
- If the user was authenticated with the secure name password contract before making an Intersite Transfer Service request to SP_A, the identity provider will not ask for the authentication.
- If the user is not authenticated while making an Intersite Transfer Service request to SP_A, the identity provider will step up to the name password form authentication.

The following diagram illustrates the workflow:



Workflow:

1. User tries to authenticate in the identity provider.
2. User is prompted to authentication using the Name Password Basic contract.
3. User enters the credentials.
4. The Name Password Basic contract is authenticated in the identity provider and added to the user session.

   The Name Password Basic contract is the default contract in the identity provider.
5. User logs into the identity provider.
6. User makes an Intersite Transfer Service request to SP_A.
7. The identity provider prompts for the authentication using the Name Password Form contract.

8 User enters the credentials.

9 The Name Password Form contract is authenticated in the identity provider and added to the user session.

10 User is redirected to SP_A.

NOTE: For information about service provider initiated single sign-on and its example, see "Contracts Assigned to a SAML 2.0 Service Provider" on page 425.

URL Query String Parameters

The following table lists query string parameters and their descriptions:

Parameter	Description
id	While defining the Intersite Transfer Service URL, you can define an id and target for the SAML service provider being accessed. For example, if you defined the id as testsaml and TARGET as URL of service provider, the login URL is https:// <identity provider server >:<port>/nidp/saml/idpsend?id=testsaml&TARGET=<URL of service provider>. If TARGET is not specified, the login URL is https:// <identity provider server >:<port>/nidp/saml/idpsend?id=testsaml.
PID	You can use this parameter when another provider is added and an Intersite Transfer Service id is not defined. For example, the login URL in this case can be `https:// <identity provider server >:<port>/nidp/saml/idpsend?PID=<https://identity provider server>:8443/nidp/saml/metadata&TARGET=<URL of service provider>`.
target	Specifies the idpsend URL with a contract ID. If authentication with a particular contract is enabled in the Intersite Transfer Service URL, you are redirected to the default target URL. Use the following format: http(s)://<$idp_host_name>/nidp/app?id=<$contract_to_be_executed>&target=http(s)://<$idp_host_name>/nidp/saml2/idpsend?id=<$saml_sp_identifier>. For example, `https://www.idp.com:8443/nidp/app?id=npbasic&target=https://www.idp.com:8443/nidp/saml2/idpsend?id=serviceprovider1`.
TARGET	Specifies URL of the final destination. Format of the URL: <identity provider URL>?PID=<entityID>&TARGET=<final_destination_URL> For example, `https://<identity provider url/nidp/saml2/idpsend?id=testsaml2&TARGET=http://www.google.com`. Here `http://www.google.com` is the destination.

Configuring Single Sign-On to Specific Applications

You can configure single sign-on through Access Gateway and through federation protocols. This section discusses how to configure single sign-on for specific applications through Access Gateway.

For more information about how to configure single sign-on through federation, see “Configuring Federated Authentication” on page 372.

- “Configuring Protected Resource for a SharePoint Server” on page 174
- “Configuring a Protected Resource for a SharePoint Server with an ADFS Server” on page 174
- “Configuring a Protected Resource for Outlook Web Access” on page 177
- “Configuring a Protected Resource for a Novell Vibe 3.3 Server” on page 180
- “Configuring Access to the Filr Site through Access Manager” on page 185

Configuring Protected Resource for a SharePoint Server

You can protect a SharePoint server as a domain-based or a path-based multi-homing resource on the Access Gateway. When you protect a SharePoint server on Access Gateway, you might see issues with rewriting if the published DNS name is not the same as the DNS name of the original server. Also, if you access SharePoint folder by using non-browser clients such as Microsoft Network Place, Nautilus in SUSE Linux Enterprise Server (SLES), or the MAC finder, you might see issues because these WebDAV clients do not support 302 redirection for authentication. You must modify the authentication procedure to prevent redirection on initial authentication or redirection to Identity Server when the user session expires.

For more information about how to configure a protected resource for a SharePoint server, see “Protecting SharePoint 2010” in the *NetIQ Access Manager 4.2 Best Practices Guide*.

Configuring a Protected Resource for a SharePoint Server with an ADFS Server

If your SharePoint server is configured to use an ADFS server and you want to create a protected resource for the SharePoint server, you need to configure the following Access Manager features. The instructions assume that you have a functioning SharePoint server and a functioning Access Manager system:

- “Configuring a Custom Contract” on page 174
- “Creating a Reverse Proxy Service” on page 175
- “Configuring Multiple Protected Resources” on page 176

Configuring a Custom Contract

ADFS requires a different format for a contract URI than the format used in the default contracts. It expects the URI to conform to the format of a URL. You need to create a custom contract.

1. In the Administration Console, click **Devices > Identity Servers > Servers > Edit > Local > Contracts**
2. Click **New**, then fill in the following fields:

   Display name: Specifies the name of the authentication contract.

URI: Specifies a value that uniquely identifies the contract from all other contracts. No spaces can exist in the URI field. For SharePoint, specify the following format for the URI:

`https://<baseurl>/name/password/uri`

Replace <*baseurl*> with the base URL of your Identity Server. If the DNS name of your Identity Server is `idp-50.amlab.net`, the URI would have the following format:

`https://idp-50.amlab.net:8443/nidp/name/password/uri`

Methods and Available Methods: Move a name/password method to the **Methods** list. We recommend **Secure Name/Password - Basic**, but you can use **Name/Password - Basic**.

Do not configure a password expiration servlet. This contract is going to be used with non-redirected login, which prevents all redirection, including redirection to a password expiration service.

For more information about other options, see "Configuring Authentication Contracts" on page 288.

3 Click **Next**.

4 Configure a card for the contract by filling in the following:

Text: Specify the text that is displayed on the card to the user.

Image: Specify the image to be displayed on the card. To use an existing image, select an image from the drop-down list. To add an image to the list, click **Select local image**.

Show Card: Determine whether the card is shown to the user, which allows the user to select and use the card for authentication. If this option is not selected, the card is only used when a service provider makes a request for the card.

5 Click **Finish**, then **OK**.

6 Update the Identity Server and the Access Gateway.

7 Continue with "Creating a Reverse Proxy Service" on page 175.

Creating a Reverse Proxy Service

1 In the Administration Console, click **Devices** > **Access Gateways** > **Edit** > **[Name of Reverse Proxy]**.

2 In the **Proxy Service List** section, click **New**.

3 Fill in the following fields:

Proxy Service Name: Specify a display name for the proxy service that the Administration Console uses for its interfaces.

Multi-Homing Type: Select **Domain-Based** as the multi-homing method that the Access Gateway must use to identify this proxy service.

Published DNS Name: Specify the DNS name you want the public to use to access the SharePoint server. This DNS name must resolve to the IP address you set up as the listening address.

If the DNS name of the reverse proxy is the same as the DNS name of the SharePoint server, no rewriting configuration is required. If they are different, there is a high probability that the application will respond incorrectly to user requests.

Web Server IP Address: Specify the IP address of the IIS Web server with the SharePoint server.

Host Header: Select the **Web Server Host Name** option.

Web Server Host Name: Specify the DNS name of the SharePoint server that the Access Gateway must forward to the Web server.

For more information about how to create a reverse proxy, see “Managing Reverse Proxies and Authentication” on page 94

4 Click **OK**.

5 Continue with “Configuring Multiple Protected Resources” on page 176.

Configuring Multiple Protected Resources

If your SharePoint server has been configured for multiple domains, you need to create three protected resources to enable single sign-on. The server has two ways to access the home page. You need to create a protected resource for each of these paths, and then a protected resource for the other pages. These protected resources must have a configuration similar to the following:

SharePoint Page	URL Path	Contract	Authentication Procedure
home page	default.aspx	custom	Normal
root	/	custom	Normal
all others	/*	custom	Non-redirected login

For single sign-on, all the protected resources need to specify the same contract. When assigning the contract for the /* resource, the contract needs to be configured to use non-redirected login for its authentication procedure. When a user first accesses the SharePoint server, the users are directed either to the home page or the root of the server. From either of these locations, the users can be redirected to the Identity Server for authentication. After the users have authenticated and the SharePoint server requests authentication for access to any of the other pages, these pages need to be configured to use non-redirected login.

1 In the **Proxy Service List**, click the name of the Proxy Service you created, then click **Protected Resources**.

2 To create a protected resource for the home page:

2a In the **Protected Resource List**, click **New**, specify a name such as `homepage`, then click **OK**.

2b For the home page of the SharePoint server, specify the following values:

Authentication Procedure: Select the custom contract you created.

URL Path: Click `/*` and replace it with `default.aspx`, then click **OK** twice.

3 To create a protected resource for the root page:

3a In the **Protected Resource List**, click **New**, specify a name such as `root`, then click **OK**.

3b For the root of the SharePoint server, specify the following values:

Authentication Procedure: Select the custom contract you created.

URL Path: Click `/*` and remove the asterisk, then click **OK** twice.

4 To create a protected resource for all other pages:

4a In the **Protected Resource List**, click **New**, specify a name such as `allothers`, then click **OK**.

4b For all other pages of the SharePoint server, specify the following values:

Authentication Procedure: Select the custom contract you created.

URL Path: Leave the default value.

4c Click the **Edit Authentication Procedures** icon on the **Authentication Procedure** line.

4d Click the name of your custom contract, then fill in the following:

Non-Redirected Login: Select this option.

Realm: Specify a name that your users associate with the SharePoint server. This name is displayed when the user needs to reauthenticate.

For more information about this feature, see "Configuring an Authentication Procedure for Non-Redirected Login" on page 105.

5 Click **OK** three times.

In the **Protected Resource List**, you must have three protected resources that use the same Authentication Procedure.

For information about configuring protected resources, see "Setting Up a Protected Resource" on page 103.

6 Click **Access Gateways**, then update the Access Gateway.

7 (Conditional) If you have limited your users to one session, modify this limitation:

7a Click **Devices** > **Identity Servers** > **Edit.**

7b Increase the value of the **Limit user sessions** option.

7c Click **OK**, then update the Identity Server.

Configuring a Protected Resource for Outlook Web Access

If you want to protect your Outlook Web Access server with the Access Gateway, you need to configure the following Access Manager features. The instructions assume that you have a functioning Outlook Web Access server and a functioning Access Manager system:

- "Configuring a Protected Resource for Outlook Web Access" on page 177
- "Configuring an Authentication Procedure" on page 178
- "Configuring a Rewriter Profile" on page 179
- "Configuring Identity Injection" on page 179
- "Configuring Form Fill" on page 180

Configuring a Protected Resource for Outlook Web Access

The following instructions assume that you have a basic setup with at least one reverse proxy and proxy service. If you don't have this basic setup, see "Managing Reverse Proxies and Authentication" on page 94 and complete a basic setup before continuing.

1 In the Administration Console, click **Devices** > **Access Gateways** > **Edit** > **[Name of Reverse Proxy]**.

2 In the **Proxy Service List** section, click **New**.

3 Specify a name for the proxy service, then click **OK**.

4 Click the newly added proxy service. Fill in the fields:

Proxy Service Name: Specify a display name for the proxy service, which the Administration Console uses for its interfaces.

Published DNS Name: Specify the DNS name you want the public to use to access your site. This DNS name must resolve to the IP address you set up as the listening address.

Multi-Homing Type: Select the multi-homing method that the Access Gateway must use to identify this proxy service.

Web Server IP Address: Specify the IP address of the IIS Web server.

Host Header: Select the **Web Server Host Name** option.

Web Server Host Name: Specify the DNS name of the Outlook Web Access server that the Access Gateway must forward to the Web server.

5 Click **OK**.

6 Continue with "Configuring an Authentication Procedure" on page 178.

Configuring an Authentication Procedure

1 Click **Access Gateways** > **Edit** > **[Name of Reverse Proxy]** > **[Name of Proxy Service]** > **Protected Resources**.

2 Click **New**, then specify a display name for the resource.

3 (Optional) Specify a description for the protected resource. You can use it to briefly describe the purpose for protecting this resource.

4 Select an authentication contract. If you want to enable non-redirected login, select **Name/Password - Basic** as the authentication contract.

5 (Optional) If you want to enable non-redirected login, click the **Edit Authentication Procedure** icon, then click the contract that you have added to specify the following information:

Non-Redirected Login: Select the option to enable non-redirected login.

Realm: Specify the security realm configured for the IIS server running the Outlook Web Access server.

To check the security realm configured for the IIS server, open the IIS Administration Console, right-click the Outlook Web Access Server the Access Gateway is protecting, then select **Properties**. The **Directory Security** tab contains the **Security realm** field.

6 Create protected resource:

6a In the **Protected Resource List**, click **New**, specify a name such as `root`, then click **OK**.

6b Specify the following values:

Authentication Procedure: Select the contract you created.

URL Path: Make sure that `/*` is selected. If you have configured Outlook Web Access as a path-based service, then click the URL path and add the path name of the service. For example, `/owa/*`, where owa is the path name.

Click **OK** twice.

7 Create a second protected resource:

7a In the **Protected Resource List**, click **New**, specify a unique name, then click **OK**.

7b Specify the following values:

Authentication Procedure: Do not select any authentication procedure because the URL path is a public resource.

URL Path: Specify `/exchweb/*as the URL path`. If you have configured Outlook Web Access as a path-based service, click the URL path and add the path name of the service. For example, `/owa/exchweb/*`, where `owa` is the path name.

Click **OK** twice.

8 Click **OK**.

9 In the **Protected Resource List**, ensure that the protected resource you created is enabled.

10 If you want to enable single sign-on, then configure Identity Injection or Form Fill policy, depending on the Outlook Web Access configuration. For more information, see “Configuring Identity Injection” on page 179.

11 Continue with “Configuring a Rewriter Profile” on page 179.

Configuring a Rewriter Profile

1 In the Administration Console, click **Devices** > **Access Gateways** > **Edit** > **[Name of Reverse Proxy]** > **[Name of Proxy Service]** > **HTML Rewriting**.

2 Click **New** in the **HTML Rewriter Profile List**.

3 Configure a Word profile:

- **3a** Specify a name for the profile, select **Word** as the search boundary, then click **OK**.
- **3b** Click **New** in the **Variable or Attribute Name to Search for Is** section, then specify `value`.
- **3c** Click **OK**.
- **3d** Select **Rewrite Inbound Query String Data**.
- **3e** Select **Rewrite Inbound Post Data.**
- **3f** Select **Rewrite Inbound Headers**.
- **3g** Ensure that **Enable Rewrite Actions** remains selected.

4 (Optional) If you have configured the path-based multi-homing service, do the following:

- **4a** Add the following content types for the **And Document Content-Type Header Is** option in the Word profile:
  - text/x-component
  - extension/htc
- **4b** Configure the following options for **Strings to Search for Is**:
  - Specify **Search** as `/exchange` and **Replace With** as `$path/exchange`
  - Specify **Search** as `/exchweb` and **Replace With** as `$path/exchweb`

5 To save your changes to browser cache, click **OK**.

6 Use the up-arrow button to move your profile to the top of the **HTML Rewriter Profile List**.

7 To apply your changes, click the **Access Gateways** link, then click **Update** > **OK**.

Configuring Identity Injection

You must configure an Identity Injection policy to enable single sign-on with the Outlook Web Access server that has basic authentication configured. This Identity Injection policy must be configured to inject an authentication header. For information about creating this policy, see “Configuring an Authentication Header Policy” on page 694.

Configuring Form Fill

You can configure a Form Fill policy to prepopulate fields in the form when you log into the Outlook Web Access first time and then save the information in the completed form to the config store for subsequent logins. For information about how to create this policy, see Chapter , “Form Fill Policies,” on page 711.

Enabling the **Auto Submit** option requires additional entries apart from the username and password fields. To enable the **Auto Submit** option:

1. In the Administration Console, click **Policies** > **Policies** > <*Policy Name*>.
2. In the Edit Policy page, add the following details under **Fill Options**:

Input Field Name	Input Field Type	Input Field Value	Data Conversion
destination	Hidden	String Constant : http://<webserver DNS Name/owa> (when Web server is configured for http.) String Constant : https://<webserver DNS Name/owa> (when Web server is configured for https.)	None
flags	Hidden	String Constant : 0	None
forcedownlevel	Hidden	String Constant : 0	None
isUt8	Hidden	String Constant : 1	None
trusted	Radio Button	String Constant : 0	None

3. Under the **Submit Options** section, select the **Enable JavaScript Handling** check box.
4. Enter `document.cookie="PBack=0; path=/"` in the **Statements to Execute on Submit** field.
5. Click **OK** and apply the changes.

Configuring a Protected Resource for a Novell Vibe 3.3 Server

The following sections explain how to configure the Access Gateway with a domain-base multi-homing service. The instructions assume that you have a functioning Novell Vibe 3.3 server on Linux and a functioning Access Manager system with a reverse proxy configured for SSL communication between the browsers and the Access Gateway.

The Novell Vibe server needs to be configured to trust the Access Gateway to allow single sign-on with Identity Injection and to provide simultaneous logout. You also need to create an Access Gateway proxy service and configure it.

- “Configuring the Novell Vibe Server to Trust the Access Gateway” on page 181
- “Configuring a Domain-Based Multi-Homing Service for Novell Vibe” on page 182
- “Creating a Pin List” on page 185

For information about other possible Access Gateway configurations, see “Teaming 2.0: Integrating with Linux Access Gateway”.

Configuring the Novell Vibe Server to Trust the Access Gateway

To use Novell Vibe as a protected resource of an Access Gateway and to use Identity Injection for single sign-on, the Teaming server needs a trusted relationship with the Access Gateway. With a trusted relationship, the Teaming server can process the authorization header credentials. The Teaming server accepts only a simple username (such as user1) and password in the authorization header.

This section explains how to set up the trusted relationship and how to enable simultaneous logout, so that when the user logs out of Teaming, the user is also logged out of the Access Gateway.

To configure the trusted relationship:

1. Log in to the Novell Vibe server.
2. Stop the Teaming server with the following command:

   ```
   /etc/init.d/teaming stop
   ```
3. Run the `installer-teaming.linux` script.
4. Follow the prompts, then select **Reconfigure settings**.
5. Follow the prompts, then select **Advanced installation**.
6. Follow the prompts, selecting the defaults until the **Enable Access Gateway** option appears, then type `Yes`.
7. In the **Access Gateway address(es)** section, include the IP address of the Access Gateway that is used for the connection to the Teaming server.

   If the Access Gateway is part of a cluster, add the IP address for each cluster member. Wildcards such as 164.99.*.* are allowed.

   When you specify IP addresses in this option, Novell Vibe logins are allowed only from the specified addresses. Also, if authorization header credentials are not present or are incorrect, the user is prompted for login by using Basic Authentication.
8. When prompted for the Logout URL, specify the URL of the published DNS name of the proxy service plus `/AGLogout`.

   For example, if the published DNS name of the proxy service is vibe`.doc.provo.novell.com`, specify the following URL:

   ```
   https://Vibe.doc.provo.novell.com/AGLogout
   ```
9. When you are prompted to use the Access Gateway for WebDAV connections, specify `No`.
10. Follow the prompts to complete the reconfiguration process.
11. Start the Vibe server with the following command:

    ```
    /etc/init.d/teaming start
    ```
12. Continue with “Configuring a Domain-Based Multi-Homing Service for Novell Vibe” on page 182.

Configuring a Domain-Based Multi-Homing Service for Novell Vibe

The following instructions describe how to set up a domain-based service to protect the Novell Vibe server. In this example, the published DNS name of the service is `Vibe.doc.provo.novell.com`. Users would access the Vibe server with a URL similar to `http://Vibe.doc.provo.novell.com`.

To configure a domain-based service for Vibe, complete the following tasks:

- “Configuring the Domain-Based Proxy Service” on page 182
- “Configuring Protected Resources” on page 183
- “Configuring a Rewriter Profile” on page 184

Configuring the Domain-Based Proxy Service

You must create a new reverse proxy before you configure the domain-based proxy service. Configure the Vibe domain as the primary proxy service and enable SSL between browser and the Access Gateway. For more information about how to create a new reverse proxy, see “Creating a Proxy Service” on page 95.

1. In the Administration Console, click **Devices** > **Access Gateways** > **Edit** > **[Name of Reverse Proxy]**.
2. In the **Reverse Proxy List**, click **New**, then specify the following details:

   Proxy Service Name: Specify a display name for the proxy service that the Administration Console uses for its interfaces.

   Multi-Homing Type: Select **Domain-Based**.

   Published DNS Name: Specify the DNS name you want the public to use to access your site. This DNS name must resolve to the IP address you set up as the listening address. For example, `vibe.doc.provo.novell.com`.

   Web Server IP Address: Specify the IP address of the Vibe server.

   Host Header: Select the **Forward Received Host Name** option.

   Web Server Host Name: Specify the DNS name of the Vibe server.
3. Click **OK**.
4. Click the newly added proxy service, then select the **Web Servers** tab.
5. Change the **Connect Port** to 8080.

   If the Novell Vibe server has port forwarding enabled, you do not need to change from the default port 80.
6. Click **TCP Connect Options**.
7. Change the value of **Data Read Timeout** option to 300 seconds.

   This longer timeout is needed for file uploads.
8. Click **OK**.
9. Continue with “Configuring Protected Resources” on page 183.

Configuring Protected Resources

You must configure an Identity Injection policy to enable single sign-on with the Novell Vibe server. This Identity Injection policy must be configured to inject the authentication credentials into the authorization headers.

1. In the Administration Console, click **Policies > Policies**.
2. Select the policy container, then click **New**.
3. Specify a name for the policy, select **Access Gateway: Identity Injection** for the type, then click **OK**.
4. (Optional) Specify a description for the injection policy. This is useful if you plan to create multiple policies to be used by multiple resources.
5. In the **Actions** section, click **New**, then select **Inject into Authentication Header**.
6. Fill in the following fields:

   User Name: Select **Credential Profile** > **LDAP User Name**.

   Password: Select **Credential Profile** > **LDAP Password**.
7. Click **OK** twice.
8. Click **Apply Changes**.

   For more information about how to create such a policy, see “Configuring an Authentication Header Policy” on page 694.

   Assign this policy to the protected resources. You need to create two protected resources, one for HTML content and one for WebDAV and AJAX content.
9. Click **Access Gateways** > **Edit** > **[Name of Reverse Proxy]** > **[Name of Proxy Service] > Protected Resources**.
10. Create a protected resource for HTML content:
    - **10a** In the **Protected Resource List**, click **New**, specify a name, then click **OK**.
    - **10b** (Optional) Specify a description for the protected resource. You can use it to briefly describe the purpose for protecting this resource.
    - **10c** Specify a value for **Authentication Procedure.** For example, select the **Secure Name/Password - Form** contract.
    - **10d** In the URL Path List, remove the /* path and add the following two paths:

      ```
      /teaming/*
      ```

      ```
      /ssf/*
      ```
    - **10e** Click **OK**.
11. Create a protected resource for WebDAV and AJAX content:
    - **11a** In the **Protected Resource List**, click **New**, specify a unique name, then click **OK**.
    - **11b** (Optional) Specify a description for the protected resource. You can use it to briefly describe the purpose for protecting this resource.
    - **11c** Click the **Edit Authentication Procedure** icon.
    - **11d** In **Authentication Procedure List**, click **New**, specify a name, then click **OK**.
    - **11e** Specify details in the following fields:

      Contract: Select the **Secure Name/Password - Form** contract, which is same contract that you selected for the HTML content protected resource.

      Non-Redirected Login: Select this option.

Realm: Specify a name that you want to use for the Teaming server. This name does not correspond to a Vibe configuration option. It appears when the user is prompted for credentials.

Redirect to Identity Server When No Authentication Header is Provided: Deselect this option.

11f Click **OK** twice.

11g For the Authentication Procedure, select the procedure you just created.

11h In the **URL Path List**, remove the `/*` path and add the following paths:

```
/ssfs/*
/ssf/rss/*
/ssf/atom/*
/ssf/ical/*
/ssf/ws/*
/ssr/*
/rest/*
```

The `/ssfs/*` path is for WebDAV content and the `/ssf/rss/*` path enables non-redirected login for RSS reader connections.

11i Click **OK**.

12 In the **Protected Resource List**, ensure that the protected resources you created are enabled.

13 To apply your changes, click **Devices** > **Access Gateways**, then click **Update**.

14 Continue with "Configuring a Rewriter Profile" on page 179.

Configuring a Rewriter Profile

1. In the Administration Console, click **Devices** > **Access Gateways** > **Edit** > **[Name of Reverse Proxy]** > **[Name of Proxy Service]** > **HTML Rewriting**.
2. In **HTML Rewriter Profile List**, click **New**.
3. Specify a name for the profile, select **Word** as the search boundary, then click **OK**.
4. In the **And Document Content-Type Header Is** section, click **New**, then specify the following type:

   ```
   application/rss+xml
   ```

5. In the **Variable or Attribute Name to Search for Is** section, click **New**, then specify the following as the variable to search for:

   ```
   value
   ```

6. Click **OK**.
7. Ensure that **Enable Rewrite Actions** remains selected.
8. Click **OK**.
9. In **HTML Rewriter Profile List**, move the Word profile you created to be the first profile in the list, and move the default profile to be the second profile in the list.
10. Click **OK**.
11. To apply your changes, click **Devices** > **Access Gateways**, **Update**.
12. Continue with "Creating a Pin List" on page 185.

NOTE: If Vibe is configured to send the binary content in the JSON format, you must disable the HTML Rewriter to prevent errors.

Creating a Pin List

Configure the Access Gateway to bypass the published URL of the proxy service:

1. In the Administration Console, click **Devices** > **Access Gateways** > **Edit.**
2. Click **Pin List** in the configuration page.
3. Click **New**, then specify the published DNS name of the proxy service. For example, `vibe.doc.provo.novell.com`.
4. Select **Bypass** as the Pin type.
5. Click **OK**.
6. To save the configuration changes, click **Devices** > **Access Gateways**, then click **Update**.

NOTE: If you do not want Access Manager to cache site information, do not create a Pin List. Instead, you must configure Access Manager to forward cache control headers to the browser. This is the recommended configuration for Novell Vibe. For information about how to forward cache control headers to the browser, see “Controlling Browser Caching” on page 254.

Configuring Access to the Filr Site through Access Manager

For information about configuring Access Manager to configure a protected resource for a Novell Filr server, see Allowing Access Manager to configure a protected resource for a Novell Filr server (http://www.novell.com/documentation/novell-filr1/filr1_admin/data/btk7698.html)

Sample Configuration for Protecting an Application Through Access Manager Appliance

This section explains how to use Access Manager Appliance to protect the Web site illustrated in the following figure. The sample application that comes by default with the Access Manager Appliance showcases the various Access Manager features. Ensure that you remove the landing portal in the production environment. Instructions for removing this portal are given on the landing page.

This section explains how to configure the Access Manager Appliance to allow access to this first page and how to create and assign policies that protect the other pages.

The example Web pages are designed to help network administrators understand the basic concepts of Access Manager Appliance by installing and configuring a relatively simple implementation of the software. The example serves as a primer for a more comprehensive production installation of Access Manager Appliance.

- “Installation Overview and Prerequisites” on page 185
- “Accessing the Sample Web Portal” on page 187
- “Understanding the Policies Used in the Sample Portal” on page 187

Installation Overview and Prerequisites

This section discusses the concepts involved in installing Access Manager Appliance to protect the example Digital Airlines website:

- “Installation Architecture” on page 186
- “Deployment Overview” on page 187

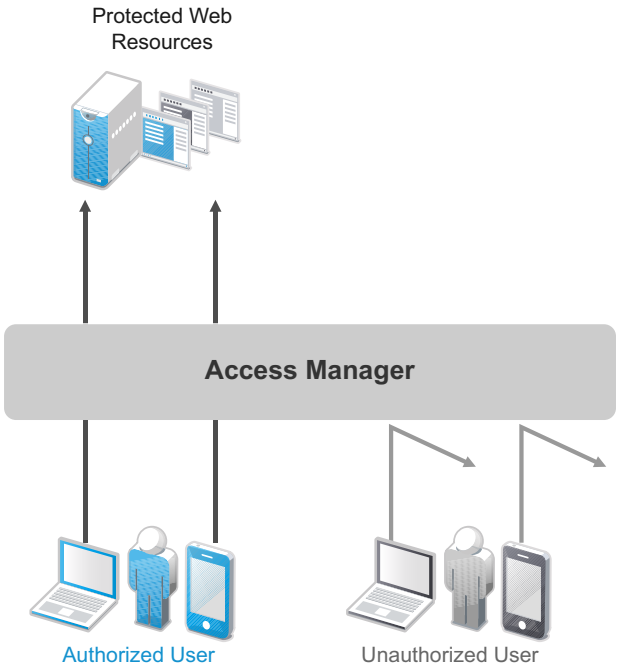
After you deploy this example, you must understand the basic features of Access Manager Appliance and know how to configure the software to protect your own Web servers and applications.

Installation Architecture

Access Manager Appliance offers a simplified deployment model. The entire product is deployed as an appliance in a single-box form factor. For more information, see Installing Access Manager Applicationin the NetIQ Access Manager Appliance 4.2 Installation and Upgrade Guide.

The primary purpose of Access Manager Appliance is to protect resources by allowing access only to users you have authorized. You can control access to Web (HTTP) resources as well as traditional server-based (non-HTTP) resources. As shown in the following illustration, those users who are authorized to use the protected resources are allowed access, while unauthorized users are denied access.

The following diagram illustrates how the sample portal is integrated with Access Manager Appliance.



Access Manager Appliance secures your protected Web resources from Internet hackers. The addresses of the servers that host the protected resources are hidden from both external and internal users. The only way to access the resources is by logging in to Access Manager Appliance with authorized credentials.

In the **Identity Server Cluster** option, the configuration assigned to the Identity Server that is the default `IDP-Cluster` is displayed. This establishes the trust relationship between the Access Gateway and the Identity Server that is used for authentication. In the **Reverse Proxy List** `NAM-RP` which is the default reverse proxy is listed.

You can see the IP address of the Access Gateway installed in the Access Gateways window. The health of the configured Access Gateway is Green. The published DNS name to access your sample web portal site, in this example uses *namapp.com*. This DNS name resolves the IP address set up as the listening address. When you edit the **Reverse Proxy / Authentication**, you can see that it is already configured.

Deployment Overview

- “Prerequisite Tasks” on page 187

Prerequisite Tasks

Before starting with the Digital Airlines example, you must perform the following tasks:

- ❒ Enable pop-ups on a Firefox browser or Internet Explorer browser for managing and configuring the Access Manager Appliance components. For information about supported version of web browsers, see NetIQ Access Manager Appliance 4.2 Installation and Upgrade Guide.
- ❒ Install the NetIQ Access Manager Appliance as described in the Installing Access Manager Applicationin the NetIQ Access Manager Appliance 4.2 Installation and Upgrade Guide.

Accessing the Sample Web Portal

You can access the sample Web portal by going to the portal Web site, in this example, *www.namapp.com/portal*. This is because the **namportal** is already configured with the published DNS name www.namapp.com and the Multi-Homing Path-Based proxy service is defined as /portal.

Protected resource details are displayed in the Protected Resource List. The portal_public is a public resource and do not have an authentication procedure. You can access this page without any credentials from the following URL:

`https://www.namapp.com/portal/` takes you to the landing page of the Web portal.

The default protected resources in this example are /portal/payinfos/* and /portal/users/* that have an associated authentication procedure. For example if you want to access the portal go to `https://www/namapp.com/portal` and click on **Sample Application** on the portal page. You will be asked for credentials. By default Access Manager creates two sample users Alice and Bob with password novell.

Understanding the Policies Used in the Sample Portal

The sample portal site is configured for authentication and role based authorization.

Access Manager Appliance uses an Identity Server Role policy to assign roles to logged in users. In the sample portal Identity Server with a policy named role_assignment Manager and Employee are defined. A user Alice is assigned with role Manager and Employee. Another user Bob is assigned with role Employee. The users of role Employee and Manager can see and edit their own as well as an employee’s basic information. Payroll information of each user is a protected information. A user who is assigned the role of Employee cannot see the pay information of other users, unless assigned the role of Manager.

Access Manager Appliance uses authorization policy to define access control. Role Based Access Control can conveniently assign a user to a particular job function or set of permissions within an enterprise. Access Manager Appliance enables you to assign roles to users, based on attributes of their identity, and then associate policies with the roles. In designing your own actual production environment, you need to decide which roles you need (such as, sales, administrative, payroll, and accounting). You can create Role policies that assign the roles to your users, and then create Authorization and Identity Injection policies that use the roles to control access.

Access Manager uses the Identity Injection policy for single sign-on to a web resource using the HTTP header, for example, HTTP authentication. There are Identity Injection policies configured with basic_auth and fillRole which are used for single sign-on to the portal. basic_auth Identity Injection

policy will inject authentication header with LDAP User DN and LDAP Password. The DN Format used is LDAP, for example, cn=alice,ou=Payroll,o=Novell. Fillrole injects the defined name and value, in this example Roles into the custom header. The main page of the sample payroll site displays the user's login name.

Access Manager uses the Form Fill policy to fill the forms from the Web server. A default Form Fill policy, fill_allowance is defined. The **Input Field Name** payinfo.allowances under **Fill Options** is defined with the value 10000. When you edit the pay info field, the **Allowances** field is automatically filled with this value. Any request without basic authentication headers and the required role will be forbidden.

You can use the sample application available to understand the roles by following the procedure below:

1. Login to the portal page for example, https://www.namapp.com/portal and click on **Sample Application**.
2. Login with the username alice. The login page is displayed with the published DNS name alice. Alice can access her pay information. If the user belongs to payroll, the **Pay Info** button is displayed on the page.
3. Click on **Employees**. Alice can access Bob's pay information because Alice is assigned the manager role. Click **show** against the DNS name, in this example, Bob and click **Pay Info**.
4. Click **pay edit** to edit the pay of the employees. The **Allowances** field is automatically filled as defined in the Form Fill policy. You can edit the pay information and save your changes.
5. Click on **New Employee** to create a new employee.

NOTE: If you login as Bob, you cannot create a new employee or access the pay information of other employees and will get a Forbidden error as Bob is not assigned a Manager role.

---

4 Setting Up an Advanced Access Manager Configuration

- “Identity Server Advanced Configuration” on page 189
- “Access Gateway Server Advance Configuration” on page 231
- “Access Gateway Content Settings” on page 252
- “Access Gateway Advanced Options” on page 259
- “Modifying Configuration Files” on page 269

Identity Server Advanced Configuration

- “Managing an Identity Server” on page 189
- “Editing Server Details” on page 192
- “Customizing The Identity Server” on page 192
- “Configuring Local Authentication” on page 271
- “Configuring SAML 2.0” on page 418
- “Configuring SAML 1.1” on page 452
- “Configuring Liberty” on page 455
- “Configuring Liberty Web Services” on page 461
- “Configuring WS Federation” on page 481
- “Configuring WS-Trust Security Token Service” on page 506
- “Configuring OAuth and OpenID Connect” on page 527

Managing an Identity Server

The Identity Servers page is the starting point for managing Identity Servers. Most often, you use this page to stop and start servers, and to assign servers to Identity Server configurations. An Identity Server cannot operate until you have assigned it to an Identity Server configuration.

1. In the Administration Console, click **Devices > Identity Servers**.
2. On the **Servers** tab, you can perform the following functions by clicking the server’s check box, then clicking any of the following options:

   Start: Starts the selected server. (See “Restarting the Identity Server” on page 191.)

   Stop: Stops the selected server. (See “Restarting the Identity Server” on page 191.)

   Refresh: Refreshes the server list.

   Actions: Enables you to perform the following task:

   - **Update Health from Server:** Performs a health check for the device.
   - **Export Configuration:** Enables you to export the Identity Server configuration to another setup. See “Exporting the Configuration Data” on page 934.

- **Import Configuration:** Enables you to import the Identity Server configuration from another setup. See “Importing the Identity Server Configuration Data” on page 936.

This page also displays links in the following columns:

Column	Description
Name	Lists Identity Server and cluster configuration names.
Status	Lists the status of each configuration.
	Current: Indicates that the server is using the latest configuration data. If you change a configuration, the system displays an **Update** or **Update All** link.
	Update: A link to update an Identity Server’s configuration data without stopping the server.
	Update All: A link displayed for cluster configurations. This lets you update all the Identity Servers in a cluster to use the latest configuration data, with options to include logging and policy settings.
	For more information about the update process, see “Updating an Identity Server Configuration” on page 191.
Health	Lists the health of each configuration and each server.
Alerts	Displays the Alerts page, where you can monitor and acknowledge server alerts.
Commands	Displays the Command Status page.
Statistics	Displays the Server Statistics page and allows you to view the server statistics. See “Monitoring Identity Server Statistics” on page 873.
Configuration	Lists the Identity Server configuration to which this server belongs.

Starting and Stopping an Identity Server Through Commands

On Linux:

Start: To start the server, run one of the following commands:

- `/etc/init.d/novell-idp start`
- `rcnovell-idp start`

Stop: To stop the server, run one of the following commands:

- `/etc/init.d/novell-idp stop`
- `rcnovell-idp stop`

On Windows:

- `net stop Tomcat7`
- `net start Tomcat7`

Updating an Identity Server Configuration

Whenever you change an Identity Server configuration, the system prompts you to update the configuration. An **Update Servers** status is displayed under the **Status** column on the Servers page. You must click **Update Servers** to update the configuration so that your changes take effect.

When you click this link, it sends a reconfigure command to all servers that use the configuration. The servers then begin the reconfiguration process. This process occurs without interruption of service to users who are currently logged in.

When you update a configuration, the system blocks inbound requests until the update is complete. The server checks for any current requests being processed. If there are such requests in process, the server waits five seconds and tests again. This process is repeated three times, waiting up to fifteen seconds for these requests to be serviced and cleared out. After this period of time, the update process begins. Any remaining requests might have errors.

During the update process, all settings are reloaded with the exception of the base URL. In most cases, user authentications are preserved; however, there are conditions during which some sessions are automatically timed out. These conditions are:

- A user logged in via an authentication contract that is no longer valid. This occurs if an administrator removes a contract or changes the URI that is used to identify it.
- A user logged in to a user store that is no longer valid. This occurs if you remove a user store or change its type. Changing the LDAP address to a different directory is not recommended, because the system does not detect the change.
- A user received authentication from an identity provider that is no longer trusted. This occurs if you remove a trusted identity provider or if the metadata for the provider changed.

Additionally, if you remove a service provider from an identity provider, the identity provider removes the provided authentication to that service provider. This does not cause a timeout of the session.

Changes to the SAML and Liberty protocol profiles can result in the trusted provider having outdated metadata for the Identity Server being reconfigured. This necessitates an update at the other provider and might cause unexpected behavior until that occurs.

1. In the Administration Console, click **Devices > Identity Servers**.
2. Click **Update** or **Update All**.

   These options are only available when you have made changes that require a server update.

Restarting the Identity Server

Starting and stopping an Identity Server terminates active user sessions. These users receive a prompt to log in again unless you have configured session failover (see “Configuring Session Failover” on page 55).

1. In the Administration Console, click **Devices > Identity Servers**, then select the Identity Server to stop.
2. Click **Stop**.
3. Wait for the **Command Status** to change from **Pending** to **Complete**.
4. Select the Identity Server, then click **Start**.
5. When the **Command Status** changes to **Complete**, click **Refresh**.

   The status icon of the Identity Server must turn green.

Editing Server Details

You can edit server details, such as the server name and port. You can also access the other server management tabs from this page.

1. In the Administration Console, click **Devices > Identity Servers**, then click the server name.
2. To edit the information, click **Edit**.
3. Modify the following fields as necessary:

   Name: The name of the Identity Server. Names must be alphanumeric and can include spaces, hyphens, and underscores.

   Management IP Address: The IP address of the Identity Server. Changing server IP addresses is not recommended and causes the server to stop reporting. See “Changing the IP Address of Access Manager Appliance” on page 47.

   Port: The Identity Server port used for management.

   Location: The location of the Identity Server.

   Description: A description of the Identity Server.
4. To save your changes, click **OK**. Otherwise, click **Cancel**.

Customizing The Identity Server

allows you to customize the Identity Server. The user portal page is what authenticates users to the Identity Server. This means when you customize the Identity Server you customize the login page, logout page, and error messages the users see.

This section includes the following topics:

- “Getting Started” on page 192
- “Customizing the Identity Server Login Page” on page 193
- “Customizing the Identity Server Logout Page” on page 209
- “Customizing Identity Server Messages” on page 211
- “Maintaining Customized Identity Server” on page 215
- “Sample Custom Login Pages” on page 215

Getting Started

Access Manager 4.2 contains a new default user portal. If you have customized the legacy user portal, you can retain the customized JSP pages.

To understand the differences between the two portal pages, see “Managing Direct Access to the Identity Server” on page 367.

To change simple aspects of the default user portal page see, “Changing the Branding of the User Portal Page” on page 769.

This section includes the following topics:

- “Customizing JSP Files” on page 193
- “Enabling the AppMark Portal Landing Page in Old Customized JSP Files” on page 193

Customizing JSP Files

1. Access the `/opt/novell/nids/lib/webapp/jsp` folder. There might or might not be multiple versions of a `*.jsp` file.
2. Any file that has `*_legacy.jsp` in the file name, add your customization to that file. For example, if there is a `nidp.jsp` file and a `nidp_legacy.jsp`, add your customization to the `nidp_legacy.jsp` file. Make no changes to the `nidp.jsp`.
3. If there is only one instance of a`*.jsp` file, add your customization to that file. For example, if there is only a `IdentityEditor.jsp` file, add your customization to the `IdentityEditor.jsp` file.
4. Create the `/opt/novell/nids/lib/webapp/WEB-INF/legacy` folder.

NOTE: For more information, see "Maintaining Customized JSP Files for Identity Server" in the *NetIQ Access Manager Appliance 4.2 Installation and Upgrade Guide*.

Enabling the AppMark Portal Landing Page in Old Customized JSP Files

Access the `/opt/novell/nids/lib/webapp/jsp` folder and make the following changes in `nidp_legacy.jsp` and `content_legacy.jsp` files:

1. Both `nidp_leagcy.jsp` and `content_legacy.jsp` contain the `ContentHandler` object that is accessed in the top Java section. Access `ContentHandler` by using the following method:

   ```
   ContentHandler handler = new ContentHandler(request,response);
   ```

2. Add the following Java line to determine if the control should be transferred to the AppMark Portal Landing page:

   ```
   boolean bGotoAlternateLandingPageUrl = handler.gotoAlternateLandingPageUrl();
   ```

3. If the `handler.gotoAlternateLandingPageUrl();` method returns "true", redirect the control to the AppMark Portal Landing page by using JavaScript.

   3a Find the first <script> </script> HTML tag pair in both files.

   3b In the JavaScript section, add the following lines inside the script tag:

   ```
   <% if (bGotoAlternateLandingPageUrl) { %>
           document.location = "<%=handler.getAlternateLandingPageUrl()%>";
   <%  } %>
   ```

Customizing the Identity Server Login Page

You can create custom login pages that are displayed when the user authenticates to the Identity Server. There are a multitude of reasons for customizing the login page. You might want to remove the NetIQ branding and replace it with your company's brands. You might need to authenticate users with non-default attributes (such as an e-mail address rather than a username). You also might be fronting several protected resources with an Access Gateway, and you need to create a unique login page for each resource.

When you customize the login page:

- You need to decide on the type of page to use. See "Selecting the Login Page and Modifying It" on page 194.
- You need to configure the Identity Server to display the correct login page. See "Configuring the Identity Server to Use Custom Login Pages" on page 203.

- If the custom page does not' display, you need to discover the cause. See "Troubleshooting Tips for Custom Login Pages" on page 208.
- You need to sanitize the JSP file to prevent XSS attacks. See "Preventing Cross-site Scripting Attacks" on page 792.

Modifying the Target of the User Portal: If you want to control the target when users log directly into the Identity Server, see "Specifying a Target" on page 369.

Modifying Error Pages: Both the Identity Server and the Access Gateway return error pages to the user. For information about customizing these messages and pages, see the following:

- "Customizing Identity Server Messages" on page 211
- "Customizing Error Messages and Error Pages on Access Gateway" on page 246.

Selecting the Login Page and Modifying It

You must be familiar with customizing JSP files to create a customized login page. You can use any of the following methods to produce the page:

- If you only need to customize the credentials (for example, prompt the user for an e-mail address rather than a name), you can make most of the modifications in the Administration Console. You need to add some properties to a method, create a contract from that method, and modify the prompt in the `login.jsp` file. For configuration information, see "Customizing the Default Login Page to Prompt for Different Credentials" on page 195.
- If you want to maintain the features of the 4.0 page and use its authentication cards but you want to remove the NetIQ branding, you need to modify the `nidp_legacy.jsp` file. The `nidp_legacy.jsp` file uses iframes, so the devices that your users use for authentication must also support iframes. For configuration information, see "Customizing the nidp.jsp File" on page 197.
- If you don't need the authentication cards and if the devices that your users use for authentication support iframes, you can start with the `login.jsp` file and customize it. For configuration information, see "Modifying the login.jsp File" on page 202.

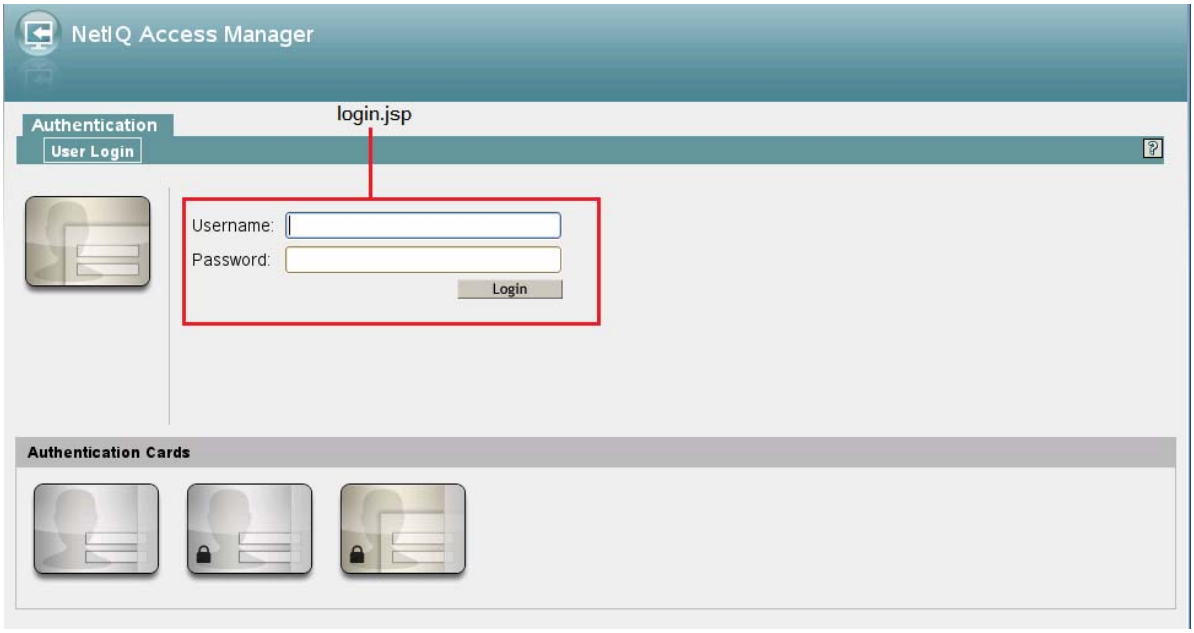
NOTE: After you have created customized login pages, you need to back them up before doing an upgrade. The upgrade process overrides any custom changes made to JSP files that use the same filename as those included with the product.

During an upgrade, you can select to restore custom login pages, but NetIQ recommends that you have your own backup of any customized files.

Customizing the Default Login Page to Prompt for Different Credentials

This section explains how to prompt the users for an identifier other than the user's name. Figure 4-1 displays the default login page with the username prompt.

Figure 4-1 *Modifying the Credential Prompts*



This section explains how to modify the content of the `login.jsp` file. If you want to modify other aspects of this page, you need to select one of the other methods.

The instructions below explain how to create a method that sets up the appropriate query so that the user can be found in the user store with an identifier other than the username (the cn attribute). The instructions then explain how to create a contract that uses this method and how to modify the `login.jsp` page so that it prompts for the appropriate identifier such as an email address instead of a username.

1 Create a method with the appropriate query:

1a In the Administration Console, click **Devices** > **Identity Servers** > **Edit** > **Local** > **Methods**.

1b Click **New**, then specify a **Display Name**.

1c In the drop-down menu for classes, select a class that is a username/password class.

1d Leave the **Identifies User** option enabled, and configure the user store option according to your needs.

1e In the **Properties** section, click **New**, then specify the following values:

Property Name: `Query`

Property Value: `(&(objectclass=person)(mail=%Ecom_User_ID%))`

This property is defined so that it queries the user store for the attribute you want to use rather than the cn attribute (in this case, the mail attribute of the person class). The `%Ecom_User_ID%` variable is the default variable name on the login page. You can change this to `%EMail_Address%` if you also change the value in your custom login page.

For more information about how to use this property, see "Query Property" on page 285.

1f In the **Properties** section, click **New**, then specify the following values:

Property Name: `JSP`

Property Value: `<filename>`

Replace *<filename>* with the name of the custom `login.jsp` page you are going to create so that the page prompts the user for an e-mail address rather than a username. This must be the filename without the JSP extension. For example, if you name your file `email_login.jsp`, then you would specify `email_login` for the property value.

1g Click **OK**.

2 Create a contract that uses this method:

2a Click **Contracts** > **New**.

2b Select the method you just created.

2c Configure the other options to fit your requirements.

For information about configuring the other options for a contract, see "Configuring Authentication Contracts" on page 288.

2d Click **OK**.

3 Update the Identity Server.

4 Copy the `login.jsp` file and rename it. The JSP files are located on the Identity Server in the following directory:

`/opt/novell/nids/lib/webapp/jsp`

5 (Conditional) If you modified the `%Ecom_User_ID%` variable, find the string in the file and replace it with your variable.

6 (Conditional) If you need to support only one language, modify the prompt in the `login.jsp` file:

6a Find the following string in the file:

```
<label><%=handler.getResource(JSPResDesc.USERNAME)%></label>
```

6b Replace it with the string you want, for example:

```
<label>Email Address:</label>
```

6c Copy the modified file to each Identity Server in the cluster.

6d Back up your customized file.

7 (Conditional) If you need to localize the prompt for multiple languages, create a custom message properties file for the login prompt. (For more information about how to create a custom message properties file, see "Customizing Messages" on page 211.)

The following steps assume you want to change the username prompt to an e-mail address prompt.

7a Find the following definition in the `com/novell/nidp/resource/jsp` directory of the unzipped `nidp.jar` file.

```
JSP.50=Username:
```

7b Add this definition to your custom properties file and modify it so that it prompts the user for an e-mail address.

```
JSP.50=Email Address:
```

7c Translate the value and add this entry to your localized custom properties files.

7d Copy the customized properties files to the `WEB-INF/classes` directory of each Identity Server in the cluster.

7e Restart Tomcat on each Identity Server using one of the following commands:

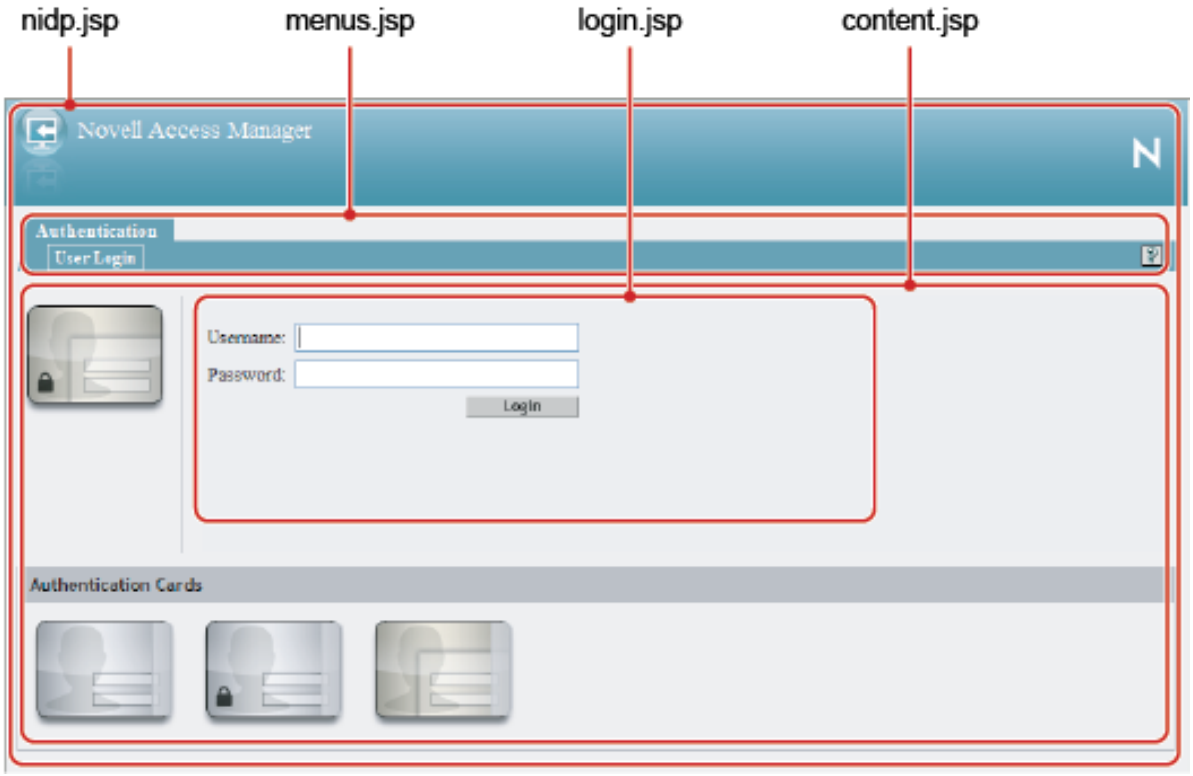
```
/etc/init.d/novell-idp restart OR
rcnovell-idp restart
```

8 To view a sample custom page with these modifications, see “Modified login.jsp File for Credential Prompts” on page 221.

Customizing the nidp.jsp File

Figure 4-2 displays the default login page provided by Access Manager. Multiple JSPs are used to create the page.

Figure 4-2 *The JSPs That Create the Login Page*



You can use the `nidp_legacy.jsp` file to customize the header with the Access Manager product name and the NetIQ logo. The `menus.jsp` file controls the **Authentication** and **User Login** tabs. The `login.jsp` file controls the credential frame with username and password. The `content.jsp` file controls what is displayed on the page, including the available authentication cards.

The following sections explain how to modify the login page that these JSPs create:

- “Logging Out of Sessions to the Access Gateway and SAML Connectors when Branding Exists in the Customized Logout Page” on page 252
- Customizing the Card Display
- Customizing the Credential Frame
- Customizing the nidp.jsp File to Customize Error Message

Rebranding the Header

1 Copy the `nidp_legacy.jsp` file and rename it. The JSP files are located on the Identity Server in the following directory:

/opt/novell/nids/lib/webapp/jsp;

2 Replace the header title that appears in the top frame (“NetIQ Access Manager” in Figure 4-2):

2a Locate the following string at the top of the file.

```
String hdrTitle = handler.getResource(JSPResDesc.PRODUCT);
```

2b Replace the value with the title you want to appear. For example:

```
String hdrTitle = "My Company";
```

Ensure that to enclose your title value with double quotes.

3 Replace the window title that appears in the browser title bar:

3a Locate the following line that appears between the <head></head> tags:

```
<title><%=handler.getResource(JSPResDesc.TITLE)%></title>;
```

3b Replace the content between the <title> and </title> tags with the title you want to appear. For example:

```
<title>My Company</title>;
```

4 Replace the Access Manager logo on the left of the header (see Figure 4-1):

4a Locate the following string:

```
String hdrImage = "AMHeader_image.png";
```

4b Replace the value in the quotes with the path and the filename of the image you want to use.

For example, if you created a /custom_images directory in the images directory, the hdrImage string would have a value similar to the following:

```
String hdrImage = "/custom_images/myapp.png"
```

4c **NOTE:** Mobile login page customization is only supported in mobile / hand-held Web browsers that support CSS 3.0 specifications. If the browser does not support CSS 3.0 specification, the default logo and header of the normal window is displayed in smaller size.

To customize the logo for a mobile view, create a new logo and name it logo_new.gif. The recommended size of the logo is 40x40 pixels.

5 Replace the NetIQ logo on the right of the header (see Figure 4-2):

5a Locate the following string:

```
String hdrLogo = "AMHeader_logo.png";
```

5b Replace the value of the hdrLogo string with the path and the filename of the image you want to use.

For example, if you created a /custom_images directory in the images directory, the hdrLogo string would have a value similar to the following:

```
String hdrLogo = "/custom_images/companylogo.png";
```

6 To change the background image for the header (which allows for variable sizing of the page):

6a Locate the following string:

```
String hdrBgndImg = "AMHeader_background.png";
```

6b Replace the value of the `hdrBgndImg` string with the path and the filename of the image you want to use. You can use a color or an image that can be repeated. The style is set to repeat it from left to right as the window expands.

For example, if you created a `/custom_images` directory in the `images` directory, the `hdrBgndImg` string would have a value similar to the following:

```
String hdrBgndImg = "/custom_images/mybackground.png";
```

6c To customize the header for a mobile view, create a new header and name it `banner_new.gif`. The recommended size of the banner is 40x800 pixels.

7 If your custom images or title do not appear in the header where you want them, you need to modify the style section.

7a Locate the following lines:

```
#header { background-image: url(<%= handler.getImage(hdrBgndImg,false)%>);
background-repeat: repeat-x; }

#logo { position: absolute; top: 0px; right: 0px; }

#title { position: absolute; font-size: 1.2em; color: white; top: 13px;
left: 55px; }
```

7b Modify the top, left, and right values.

8 To change the background colors on the page, modify the color values in the `<style>` section of the `<head>` element.

9 If you need to create multiple custom login pages, repeat Step 1 through Step 8.

10 Copy the custom login pages and the images they require to each Identity Server in the cluster.

11 Continue with one of the following tasks:

- To modify what appears in the credential frame, continue with “Customizing the Credential Frame” on page 200.
- To control the cards displayed in the Authentication Cards section, see “Customizing the Card Display” on page 199.
- To configure the Identity Server to use your custom pages, see “Adding Logic to the main.jsp File” on page 204.
- To view a sample custom page with these modifications, see “Custom nidp.jsp File with Custom Credentials” on page 224.

Customizing the Card Display

The easiest method to control what appears in the **Authentication Cards** section is not by modifying the `content.jsp` file. It is by using the **Show Card** option that appears on the definition of each card. If this option is not selected, the card does not appear in the **Authentication Cards** section. Each contract has an associated card. For information about modifying the card options, see “Configuring Authentication Contracts” on page 288.

Continue with one of the following:

- To modify what appears in the credential frame, continue with “Customizing the Credential Frame” on page 200
- To configure the Identity Server to use your custom pages, see “Adding Logic to the main.jsp File” on page 204.

Customizing the Credential Frame

The most common reason for modifying the `login.jsp` page is to prompt the users for an identifier other than the user's name. To do this, you need to create a method that sets up the appropriate query so that the user can be found in the user store with an identifier other than the username. You then need to create a contract that uses this method. You also need to modify the prompt in the `login.jsp` page to match the identifier you are prompting for.

1 Create a method with the appropriate query:

1a In the Administration Console, click **Devices** > **Identity Servers** > **Edit** > **Local** > **Methods**.

1b Click **New**, then specify a **Display Name**.

1c In the drop-down menu for classes, select a class that is a username/password class.

1d Leave the **Identifies User** option enabled, and configure the user store option according to your needs.

1e In the **Properties** section, click **New**, then specify the following values:

Property Name: `Query`

Property Value: `(&(objectclass=person)(mail=%Ecom_User_ID%))`

This property is defined so that it queries the user store for the attribute you want to use rather than the cn attribute (in this case, the mail attribute of the person class). Change `mail` to the name of the attribute in your user store that you want to use for the user identifier.

The `%Ecom_User_ID%` variable is the default variable name on the login page. You can change this to something like `%EMail_Address%` if you also change the value in your custom login page.

For more information about how to use this property, see "Query Property" on page 285.

1f In the **Properties** section, click **New**, then specify the following values:

Property Name: `JSP`

Property Value: `<filename>`

Replace <*filename*> with the name of the custom `login.jsp` page you are going to create so that the page prompts the user for an e-mail address rather than a username. This must be the filename without the JSP extension. For example, if you name your file `email_login.jsp`, then you would specify `email_login` for the property value.

1g Click **OK**.

2 Create a contract that uses this method:

2a Click **Contracts** > **New**.

2b Select the method you just created.

2c Configure the other options to fit your requirements.

If you are creating multiple custom login pages with customized credentials, you might want to use the URI to hint at which custom `login.jsp` file is used with which custom `nidp_legacy.jsp` file. For example, the following URI values have the filename of the login page followed by the name of the custom `nidp_legacy.jsp` page:

```
login1/custom1
login2/custom2
login3/custom3
```

For information about configuring the other options for a contract, see "Configuring Authentication Contracts" on page 288.

2d Update the Identity Server.

3 Copy the `login.jsp` file and rename it. The JSP files are located on the Identity Server in the following directory:

   `/opt/novell/nids/lib/webapp/jsp`

4 (Conditional) If you modified the `%Ecom_User_ID%` variable, find the string in the file and replace it with your variable.

5 (Conditional) If you need to support only one language, modify the prompt in the `login.jsp` file:

   5a Find the following string in the file:

   ```
   <label><%=handler.getResource(JSPResDesc.USERNAME)%></label>
   ```

   5b Replace it with the string you want, for example:

   ```
   <label>Email Address:</label>
   ```

   5c Copy the modified file to each Identity Server in the cluster.

   5d Back up your customized file.

6 (Conditional) If you need to localize the prompt for multiple languages, create a custom message properties file for the login prompt. (For more information about how to create a custom message properties file, see "Customizing Messages" on page 211.)

   The following steps assume you want to change the username prompt to an e-mail address prompt.

   6a Find the following definition in the `com/novell/nidp/resource/jsp` directory of the unzipped `nidp.jar` file.

   ```
   JSP.50=Username:
   ```

   6b Add this definition to your custom properties file and modify it so that it prompts the user for an e-mail address.

   ```
   JSP.50=Email Address:
   ```

   6c Translate the value and add this entry to your localized custom properties files.

   6d Copy the customized properties files to the `WEB-INF/classes` directory of each Identity Server in the cluster.

   6e Restart each Identity Server using one of the following commands:

   ```
   /etc/init.d/novell-idp restart
   ```

   ```
   rcnovell-idp restart
   ```

7 To view a sample custom page with these modifications, see "Custom nidp.jsp File with Custom Credentials" on page 224.

8 To specify which customized `nidp_legacy.jsp` to display with the contract, you must modify the `main.jsp` file. Continue with "Adding Logic to the main.jsp File" on page 204.

Customizing the nidp.jsp File to Customize Error Message

The Identity Server publishes a generic error message for error code during SAML failure such as request denied or Invalid Name ID Policy and so on. You can customize the NIDP jsp file available at `/opt/novell/nids/lib/webapp/jsp` and write an appropriate error message for either redirection or to inform the user about the issue with an appropriate message. Perform the following steps to customize error message.

NOTE: In the following example the specified code snippet is for simulating InvalidNameIDPolicy error for SAML 2.0.

1. Generate an error condition with for example, Invalid Name ID Policy.
2. Customized the nidp.jsp file and add the following code for redirection.

```
com.novell.nidp.ui.MenuHandler redirectMenuHandler;
        com.novell.nidp.NIDPMessage redirectMessage;
    String redirectCause;

        redirectMenuHandler = new MenuHandler(request, response);
        redirectMessage = redirectMenuHandler.getMessage(true);
        if (redirectMessage != null && redirectMessage instanceof
com.novell.nidp.NIDPError) {
        redirectCause = ((com.novell.nidp.NIDPError)
redirectMessage).getNIDPExceptionMsg();
        System.out.println("************** redirectCause" + redirectCause);
        if (redirectCause != null &&
redirectCause.indexOf("InvalidNameIDPolicy") != -1) {
        response.sendRedirect("http://www.novell.com");
            return;
        }
    }
```

3. Restart the Identity Server by using the `rcnovell-idp restart` command.
4. Verify that when failure occurs, SAML shows the following message in the authentication response.

```
<samlp:Status><samlp:StatusCode
Value="urn:oasis:names:tc:SAML:2.0:status:Responder"><samlp:StatusCode
Value="urn:oasis:names:tc:SAML:2.0:status:InvalidNameIDPolicy"/></
samlp:StatusCode>
```

   Due to customized `nidp_legacy.jsp` file, SAML redirects to specified location.
5. Rerun the failure and verify that instead of displaying 300101008, nidp page redirects to the specified www.novell.com location.

Modifying the login.jsp File

The `login.jsp` file gives you just the credential frame with the login prompts in an iframe. It has no branding header. If you use this page, you are responsible for writing the HTML code for the header and the branding.

1. Copy the `login.jsp` file and rename it. The JSP files are located on the Identity Server in the following directory:

   `/opt/novell/nids/lib/webapp/jsp`
2. Add the custom branding and any other content you require to the file.
3. To modify the credentials, see “Customizing the Credential Frame” on page 200.
4. Repeat Step 1 through Step 3 for each resource that requires unique branding.
5. Copy the files to each Identity Server in the cluster.
6. Back up your customized files.
7. Continue with “Using Properties to Specify the Login Page” on page 203.

Configuring the Identity Server to Use Custom Login Pages

There are two ways to configure the Identity Server to use a custom login page. You can use properties or you can modify the `main.jsp` file. Which method you can use depends upon your modifications.

- You can use properties if you created your custom page from the 3.2 or later `login.jsp` page. See "Using Properties to Specify the Login Page" on page 203.
- If you created your custom page from the `nidp_legacy.jsp` file, you cannot use properties to specify the main custom page for authentication. You must modify the `main.jsp` file. See "Adding Logic to the main.jsp File" on page 204.

Using Properties to Specify the Login Page

For each resource that needs a unique login page, you need to create an authentication method and add the JSP and MainJSP properties to the method. You then need to create a contract for each method.

The following steps assume that the custom login page is called `custom1.jsp`.

1 Create a method for a custom login page:

1a In the Administration Console, click **Devices** > **Identity Servers** > **Edit** > **Local** > **Methods**.

1b Select one of the following actions:

- If you have create a method for a Query property to be used with your custom login page, click the name of the method.
- If you didn't modify the credentials on the login page, click **New**, specify a display name, select a password class, and configure a user store.

1c In the **Properties** section, click **New**, then specify the following:

Property Name: `MainJSP`

Property Value: `true`

This property indicates that you want to use a custom login page with this method. It also indicates that the custom login page contains the prompts for user credentials.

Property names and values are case sensitive.

1d Click **OK**.

1e (Conditional) If the **Properties** section does not contain a JSP property, click **New**, specify the following:

Property Name: `JSP`

Property Value: `custom1`

The property value for the JSP property is the name of the custom login file without the `JSP` extension. Replace `custom1` with the name of your custom login file. This property determines which login page is displayed when this method is used. The filename cannot contain `nidp` as part of its name.

1f Click **OK**.

For more information about setting property values, see "Specifying Common Class Properties" on page 285.

1g (Conditional) If you created multiple custom login pages, repeat Step 1b through Step 1e for each page.

2 For each method that you modified for a custom login page, create a contract:

2a Click **Contracts**, then click **New**.

2b Fill in the fields to fit the needs of the resource, but ensure that to assign the custom method as the method for the contract.

2c Click **Next**, configure a card for the contract, then click **Finish**.

3 Update the Identity Server.

4 For each resource that you have created a custom login page, assign that resource to use the contract that is configured to display the appropriate login page:

4a Click **Devices** > **Access Gateways** > **Edit** > **[Reverse Proxy Name]** > **[Proxy Service Name]** > **Protected Resources**.

4b For each protected resource that you have created a custom contract for, select the protected resource, then configure it to use the custom contract.

5 Update the Access Gateway.

6 (Conditional) If the custom page does not display correctly, see "Troubleshooting Tips for Custom Login Pages" on page 208.

Adding Logic to the main.jsp File

You can modify the `main.jsp` file and use the contract URI to specify the login page to display. The Identity Server must be running 3.2 or later to use this feature. Be aware of the following:

- The `main.jsp` file cannot be renamed, so any modifications you make to this file can be lost whenever you upgrade the Identity Server. During the upgrade, you must select to restore custom files or you must restore your modified file after the upgrade. If this is the only JSP file that you modified that uses an Identity Server name, it is probably best to manually restore this file after an upgrade.
- Modifying the `main.jsp` file requires knowledge of JSP programming and if/else statements.

Modifying the `main.jsp` file allows you to have the following type of configuration:

- You can create multiple customized `nidp_legacy.jsp` pages. For example: `custom1.jsp`, `custom2.jsp`, and `custom3.jsp`.
- You can create multiple customized `login.jsp` pages that request different login credentials. For example:

  login1.jsp: Configured to request username and password.

  login2.jsp: Configured to request username, email, and password.

  login3.jsp: Configured to request email and password.

With this type of configuration, you must create three different authentication contracts with an authentication method with a JSP property defined for each of them. These contracts require the types of values listed in the table below. The URI is defined so that it reflects the custom `login.jsp` and the custom `nidp_legacy.jps` that are used by the contract.

Contract	Configuration Details	
Contract1	URI	login1/custom1

Contract	Configuration Details	
	Method1	Configured with the following JSP property: **Property Name:** `JSP` **Property Value:** `login1` This method does not need a query property unless you are using an attribute other than the cn attribute for the username.
Contract2	URI	login2/custom2
	Method2	Configured with the following two properties: **Property Name:** `JSP` **Property Value:** `login2` **Property Name:** `Query` **Property Value:** `(&(objectclass=person)(mail=%Ecom_User_ID%)))`
Contract3	URI	login3/custom3
	Method3	Configured with the following two properties: **Property Name:** `JSP` **Property Value:** `login3` **Property Name:** `Query` **Property Value:** `(&(objectclass=person)(mail=%Ecom_User_ID%))`

The following procedure explains how to configure Access Manager to display these custom login pages with custom credentials.

1 Create a unique method for each custom `login.jps` file:

1a In the Administration Console, click **Devices** > **Identity Servers** > **Edit** > **Local** > **Methods**.

1b Click **New**, then configure the following fields:

Display name: Specify a name for the method. You might want to use a name that indicates which login page is assigned to this method.

Class: Select a name/password class.

Configure the other fields to match your requirements.

1c In the **Properties** section, add a Query property if the page uses custom credentials.

For example, to add an email address to the login prompts, add the following property:

Property Name: `Query`

Property Value: `(&(objectclass=person)(mail=%Ecom_User_ID%))`

If you are creating a method for Contract 1 in the example above (which prompts for a username and password), you do not need to add a query property unless you are using an attribute other than the cn attribute for the username.

1d In the Properties section, add a JSP property to specify which `login.jsp` file to use with this method.

For example:

Property Name: `JSP`

Property Value: `login2`

1e Click **Finish**.

1f If you have created more than one custom `login.jsp` file, repeat Step 1b through Step 1e for each page.

To configure the scenario described in this section, repeat these steps for three login pages.

2 Create a unique contract URI:

2a In the Administration Console, click **Contracts**.

2b Click **New**, then configure the following fields:

Display name: Specify a name for the contract. You might want to use a name that indicates which login page is assigned to this contract.

URI: Specify a value that uniquely identifies the contract from all other contracts. No spaces can exist in the URI field. You might want to use a name that indicates the custom login page and custom credential page, such as `login1/custom1`.

Methods and Available Methods: Select the authentication method you configured in Step 1.

2c Configure the other fields to meet your network requirements, then click **Next**.

2d Configure the authentication card, then click **Finish**.

2e (Conditional) If you have created multiple custom login pages, repeat Step 2b through Step 2d for each page.

To configure the scenario described in this section, repeat these steps for /login2/custom2 and /login3/custom3.

2f Click **OK**, then update the Identity Server.

3 Modify the `main.jsp` file:

3a Open the `main.jsp` file. The file is located in the following directory:

/opt/novell/nids/lib/webapp/jsp

3b Near the top of the file, add the following line:

```
String strContractURI = hand.getContractURI();
```

This sets the `strContractURI` variable to the value of the contract URI that is being used for authentication. These lines must look similar to the following:

```
<%
    ContentHandler hand = new ContentHandler(request,response);
    String strContractURI = hand.getContractURI();

    // Is there a JSP defined on a class definition or a method
    // definition that must be displayed as the main jsp here?
    if (handler.contractDefinesMainJSP())
    {
%>
```

3c After the `if` statement, add an `else if` statement for each contract URI you have created. For example:

```
<% }
else if(strContractURI != null && strContractURI.equals("login1/custom1"))
    {
%>
     <%@ include file="custom1.jsp" %>

<%  }
else if(strContractURI != null && strContractURI.equals("login2/custom2"))
    {
%>
        <%@ include file="custom2.jsp" %>

<% }
else if(strContractURI != null && strContractURI.equals("login3/custom3"))
    {
%>
        <%@ include file="custom3.jsp" %>
```

These `else if` statements set up three contracts for customized login pages:

- The first `else if` statement specifies the URI of the login1 contract and configures it to display the `custom1.jsp` page for authentication.
- The second `else if` statement specifies the URI of the login2 contract and configures it to display the `custom2.jsp` page for authentication.
- The third `else if` statement specifies the URI of the login3 contract and configures it to display the `custom3.jsp` page for authentication.

Your file must look similar to the following:

```
<%@ page language="java" %>
<%@ page pageEncoding="UTF-8" contentType="text/html; charset=UTF-8"%>
<%@ page import="com.novell.nidp.*" %>
<%@ page import="com.novell.nidp.resource.jsp.*" %>
<%@ page import="com.novell.nidp.ui.*" %>
<%@ page import="com.novell.nidp.common.util.*" %>
<%@ page import="com.novell.nidp.liberty.wsf.idsis.apservice.schema.*" %>

<%
    ContentHandler hand = new ContentHandler(request,response);
    String strContractURI = hand.getContractURI();

    // Is there a JSP defined on a class definition
    // or a method definition that must be displayed
    // as the main jsp here?
    if (hand.contractDefinesMainJSP())
    {
%>
        <%@ include file="mainRedirect.jsp" %>
<%  }
    else if(strContractURI != null && strContractURI.equals("login1/
custom1"))
    {
%>
     <%@ include file="custom1.jsp" %>

<%  }
else if(strContractURI != null && strContractURI.equals("login2/custom2"))
    {
```

```
%>
        <%@ include file="custom2.jsp" %>

else if(strContractURI != null && strContractURI.equals("login3/custom3"))
    {
%>
        <%@ include file="custom3.jsp" %>

<%  }    // This is the jsp used by default
    else
    {
%>
        <%@ include file="nidp.jsp" %>
<%  }  %>
```

3d Copy the modified `main.jsp` file to each Identity Server in your cluster.

4 Back up your customized files.

5 For each resource that you have created a custom login page for, assign that resource to use the contract that is configured to display the appropriate login page:

5a Click **Devices** > **Access Gateways** > **Edit** > **[Reverse Proxy Name]** > **[Proxy Service Name]** > **Protected Resources**.

5b For each protected resource that you have created a custom contract for, select the protected resource, then configure it to use the custom contract.

5c Update the Access Gateway.

6 (Conditional) If the custom page does not display correctly, see "Troubleshooting Tips for Custom Login Pages" on page 208.

Troubleshooting Tips for Custom Login Pages

If your custom login page does not display or generates an error message, use the following procedure to discover the root cause:

1 Set the **Application** option of **Component File Logger Levels** to debug, update the Identity Server, attempt to log in, then view the log file.

Check for Unable to Compile errors in the log file. If your custom page does not compile, a blank page is displayed.

2 If you receive an Unable to Find File error, verify the value of the JSP property. Ensure that the value does not contain the JSP extension as part of the filename.

3 If you see pages that you have deleted or pages where your modifications have not been implemented:

3a Open the new custom file with a text editor to ensure it has a newer date than the compiled file.

If this does not solve the problem, continue with Step 3b.

3b Delete the `nidp` directory in the Tomcat work directory on each Identity Server to recompile JSP pages.

`/opt/novell/nam/idp/work/Catalina/localhosts/nidp`

3c Restart Tomcat on each Identity Server.

Customizing the Identity Server Logout Page

You can also use the following methods to modify the Identity Server logout page:

- “Rebranding the Logout Page” on page 209
- “Replacing the Logout Page with a Custom Page” on page 209
- “Configuring for Local Rather Than Global Logout” on page 210
- “Customizing Logout Pages to Redirect Based on Parameters” on page 210

To customize the logout page when the user logs out of an Access Gateway protected resource, see “Customizing Logout Requests” on page 249. When you have both Liberty and SAML 2.0 sessions running on the Identity Server and you log out of the Access Gateway, the `logoutsuccess_legacy.jsp` page is not executed with the customization you have made to the logout page. For information about the workaround, see “Logging Out of Sessions to the Access Gateway and SAML Connectors when Branding Exists in the Customized Logout Page” on page 252.

NOTE: After customizing a JSP file, you need to sanitize the JSP file to prevent XSS attacks. See, “Preventing Cross-site Scripting Attacks” on page 792.

Rebranding the Logout Page

The branding in the header of the logout page is controlled by the branding of the `nidp_legacy.jsp` file. If you have modified this file for a customized login, the same branding appears in the logout page. For information about how to modify `nidp_legacy.jsp` for logos, titles, and colors, see “Logging Out of Sessions to the Access Gateway and SAML Connectors when Branding Exists in the Customized Logout Page” on page 252.

IMPORTANT: Save a copy of your modified `nidp_legacy.jsp` file. Every time you upgrade your Identity Server, you need to restore this file.

Replacing the Logout Page with a Custom Page

You can create your own logout page and configure the Identity Server to use it. To do this, you need to modify the `logoutSuccess_legacy.jsp` file on the Identity Server. It is located in the following directory:

```
/opt/novell/nids/lib/webapp/jsp
```

The `logoutSuccess_legacy.jsp` file is called in a frame from the `nidp_legacy.jsp` file. You can modify the file to display what you want or you can modify it to redirect the user to your custom page. One way to provide redirection is to replace the information in the <body> element of the file with something similar to the following:

```
<body>
      <script language="JavaScript">
        top.location.href='http://<hostname/path>';
      </script>
</body>
```

Replace the <*hostname/path*> string with the location of your customized logout page.

IMPORTANT: Save a copy of your modified `logoutSuccess_legacy.jsp` file. Every time you upgrade your Identity Server, you will need to restore this file.

Configuring for Local Rather Than Global Logout

By default, when the Identity Server receives a logout request, the Identity Server logs the user out of any identity providers and service providers to which the user has authenticated. If you want to modify this behavior so that the logout request logs the user out of just the Identity Server and leaves the user authenticated to identity providers and service providers, you need to add the following query string to the logout URL:

```
?local=true
```

The logout URL has the following format:

```
<Base_URL>/app/logout
```

Replace <*Base_URL*> with the base URL of your Identity Server. If the base URL of your Identity Server was `https://hhb1.provo.novell.com:8443/nidp`, your local logout URL would be the following:

```
https://hhb1.provo.novell.com:8443/nidp/app/logout?local=true
```

To modify the `logout.jsp` file so that it performs a local logout:

1. At the Identity Server, open the logout.jsp file.

   ```
   /opt/novell/nids/lib/webapp/jsp
   ```

2. Find the following line in the file:

   ```
   <form method="post" target="_top" action="<%= request.getContextPath() %>/app/
   logout">
   ```

3. To the `/app/logout` string, add `?local=true`. This modified line must look similar to the following:

   ```
   <form method="post" target="_top" action="<%= request.getContextPath() %>/app/
   logout?local=true">
   ```

4. Save the file.
5. Copy the file to each Identity Server in the cluster.
6. Back up this file.

Customizing Logout Pages to Redirect Based on Parameters

The Identity Server logout page can be customized to redirect to URLs based on parameters specified in the `logoutSuccess_legacy.jsp` file. To customize the `logoutSuccess_legacy.jsp` file to redirect to URLs, perform the following steps:

1. At the Identity Server, open the `logoutSuccess_legacy.jsp` file:

   Linux: `/opt/novell/nids/lib/webapp/jsp`

   Windows: `\Program Files (x86)\Novell\Tomcat\webapps\nidp\jsp`

2. Add the following as the last line in the `logoutSuccess_legacy.jsp` file:

   ```
   <%out.println("UIHandler-param: " + uh.getLogoutQueryStringParam("test"));%>
   ```

   Here *test* indicates name of the parameter.

3. Restart the Identity Server.
4. Specify a parameter with the logout URL. For example: `https://www.idp.com:8443/nidp/app/logout?test=NetIQ`.

   The logout page displays `UIHandler-param: NetIQ` in the logout page.

Customizing Identity Server Messages

- "Customizing Messages" on page 211
- "Customizing the Branding of the Error Page" on page 213
- "Customizing Tooltip Text for Authentication Contracts" on page 214

NOTE: After customizing a JSP file, you need to sanitize the JSP file to prevent XSS attacks. See, "Preventing Cross-site Scripting Attacks" on page 792.

Customizing Messages

1 To customize the error pages, determine whether you need one custom file or multiple files:
   - If you do not need to support multiple languages, you can create one custom file for all your customized messages.
   - If you need to support multiple languages, you need to create a custom file for each language you want to customize.

2 Create the custom properties file and name it:

   To support one language, name the file `nidp_custom_resources.properties`.

   To support multiple languages, create a `nidp_custom_resources_<le_cy>.properties` file for each supported language. Replace *<le_cy>* with the standard convention for Java Resource Bundles for the language or the language and country. For example:

   ```
   nidp_custom_resources_en_US.properties
   nidp_custom_resources_fr.properties
   nidp_custom_resources_es.properties
   ```

   If you want to support a custom messages for a language and a country and for just the language, you must create two files. For example:

   ```
   nidp_custom_resources_es_VE.properties
   nidp_custom_resources_es.properties
   ```

3 Copy the `nidp.jar` file to a working area. This file is located in the following directory:

   `/opt/novell/nids/lib/webapp/WEB-INF/lib`

4 Unzip the `nidp.jar` file in your working directory.

5 In your working directory, locate the `.properties` files in the following directories.

   ```
   com/novell/nidp/resource/strings
   com/novell/nidp/resource/logging
   com/novell/nidp/resource/jsp
   com/novell/nidp/resource/jcc
   com/novell/nidp/resource/noxlate
   com/novell/nidp/liberty/wsf/idsis/ppservice/model
   com/novell/nidp/liberty/wsf/idsis/epservice/model
   com/novell/nidp/liberty/wsf/idsis/opservice/model
   com/novell/nidp/liberty/wsf/idsis/apservice/model
   com/novell/nidp/liberty/wsf/interaction
   com/novell/nidp/liberty/wsf/idsis/ssservice/model
   com/novell/nidp/servlets/handler/identityeditor
   com/novell/nidp/servlets/handler/identityaccesseditor
   com/novell/nidp/liberty/wsf/idsis/model
   com/novell/nidp/liberty/wsf/idsis/authority/ldap/attribute/plugins/resources
   com/novell/nidp/liberty/wsf/idsis/ldapservice/model
   ```

The properties files that have been localized contain the messages that end users might see. The properties files that have not been localized contain messages that the end users must not see.

6 Locate the messages you want to customize and copy them to your custom file.

All the messages you want to customize are placed in this file, even though they come from different properties files. Your file must look similar to the following if you selected to customize messages from the `nidp_resources_en_US.properties` file and the `SSModelResources_en_US.properties` file. For example:

```
NIDPMAIN.100=An Identity Provider response was received that failed to
authenticate this session.
NIDPMAIN.101=A request for identity federation could not be completed.
NIDPMAIN.102=A request for identity federation termination could not be
completed.

SS.WKSLdapCreds = LDAP Credentials
SS.WKSELdapCredsUserName = LDAP User Name
SS.WKSELdapCredsUserDN = LDAP User DN
SS.WKSELdapCredsUserPassword = LDAP Password
SS.WKSX509Creds = X509 Credentials
```

7 (Conditional) If you are supporting multiple languages, copy the messages to each custom language file.

8 Replace the messages in the file with your custom messages.

Replace the string after the equals (=) sign with your translated or customized message.

If you are using double-byte characters, the characters need to be in Unicode, hexadecimal format with a `\u` prefix. For example: `\u5c71`.

9 Save the file.

10 Copy the custom properties file to the following directory on all Identity Servers in the cluster:

`/opt/novell/nam/idp/webapps/nidp/WEB-INF/classes`

11 (Optional) To enable messages about the loading of the custom properties files, enable debug logging:

11a In the Administration Console, click **Devices** > **Identity Servers** > **Edit** > **Logging**.

11b In the **Component File Logger Levels** section, select **Debug** level for **Application**.

11c Click **OK**, then update the Identity Server.

12 Restart Tomcat.

`/etc/init.d/novell-idp restart Or`

`rcnovell-idp restart`

13 (Optional) To verify the loading of the custom properties files:

13a View the log file by clicking **Auditing** > **General Logging**.

13b Search for messages similar to the following in the `catalina.out` or `stdout.log` file:

```
The named Custom Properties File was loaded and will be used:

Custom Properties File successfully loaded! Name: <Custom Properties
FileName>

An error occurred loading a specific Custom Properties File. Loading of
other Custom Properties Files will continue.

<Error Description>, Attempting to load Custom Properties File! Name:
<Custom Properties FileName>

The locale specifier in the Custom Properties File filename could not be
successfully parsed into a valid locale. Loading of other Custom Properties
Files will continue.

Custom Properties File load failed. Could not determine correct locale!
Name: <Custom Properties FileName>

A general error occurred loading Custom Properties Files. Loading will stop
and all un-loaded Custom Properties Files will not be loaded.

<Error Description>, Attempting to load Custom Properties Files!
```

To create custom error pages for the Access Gateway, see "Customizing Error Messages and Error Pages on Access Gateway" on page 246.

Customizing the Branding of the Error Page

The error page (`err_legacy.jsp`) is returned when the Identity Server encounters an error with the following message:

```
Error: Unable to authenticate, (300101014-esp-01E79F6000B87D4E8)
```

The file is located in the following directory.

```
/opt/novell/nids/lib/webapp/jsp
```

IMPORTANT: After you have customized this page, you need to ensure you back up this page before doing an upgrade. The upgrade process overrides any custom changes made to the `err_legacy.jsp` page.

For information about customizing the error message, see "Customizing Messages" on page 211.

You can customize the following items:

- The window title and the display title. See "Customizing the Titles" on page 213.
- The header image and the Novell logo. See "Customizing the Images" on page 214.
- Background colors. See "Customizing the Colors" on page 214.

Customizing the Titles

The window title appears in the browser title bar. To replace this text, open the `err_legacy.jsp` file and locate the following text that appears between the `<head></head>` tags:

```
<title><%=handler.getResource(JSPResDesc.TITLE)%></title>
```

Replace the content between the `<title>` and `</title>` tags with the title you want to appear. For example:

```
<title>My Company</title>
```

The display title is the title that appears in the top frame of the page. Locate the following text that appears in the `<body>` of the page:

```
<div id="title"><%=handler.getResource(JSPResDesc.PRODUCT)%></div>
```

Replace the content between the `<div id="title">` and `</div>` with the title you want to appear. For example:

```
<div id="title">My Company</div>
```

Customizing the Images

To replace the header image, open the `err_legacy.jsp` file and locate the following text in the body of the file.

```
<div><img src="/nesp/images/AccessMan_Login_Head.png"></div>
```

Replace the value of the `src` attribute with the path and filename of the image you want to use.

To replace the Novell logo image, locate the following text in the body of the file.

```
<div id="logo"><img src="/nesp/images/AccessMan31_Nlogo.png"></div>
```

Replace the value of the `src` attribute with the path and filename of the image you want to use.

Customizing the Colors

To change the background colors on the page, modify the color values in the `<style>` section of the `<head>`.

Customizing Tooltip Text for Authentication Contracts

The strings that the users see when they mouse over the cards for authentication contracts can be customized. If you need to support only one language, modify the text in the Administration Console.

1. In the Administration Console, click **Devices** > **Identity Servers** > **Edit** > **Local** > **Contracts**.
2. Click the name of a contract, then click **Authentication Card**.
3. Replace the English text in the **Text** option with the required language, then click **OK**.
4. Repeat Step 2 and Step 3 for each contract in the list.
5. Click **OK**, then update the Identity Server.

If you need to support multiple languages, you need to localize the tooltips. The nidsCardText attribute of the nidsAuthLocalContract object needs to be changed to a resource ID. The following procedure explains how to do this in the Administration Console. You can also use an LDAP browser.

1. In the Administration Console, click **Devices** > **Identity Servers** > **Edit** > **Local** > **Contracts**.
2. Click the name of a contract, then click **Authentication Card**.
3. Replace the text in the **Text** option with a resource ID.

   For example, replace `Name/Password - Form` with `CUSTOM_NamePwdFormToolTip`.
4. Click **OK**.

5 Repeat Step 2 through Step 4 for each contract in the list.

6 Click **OK**, then update the Identity Server.

7 Use custom string resource files to define the localized strings:

7a Change to the `WEB-INF/classes` directory.

7b For each supported language, create a properties file. For example:

```
nidp_custom_resources_fr.properties
nidp_custom_resources_es.properties
```

If you have already created these files for custom messages (see "Customizing Messages" on page 211), use the existing files.

7c For each resource ID you have created, add an entry that contains the resource ID and the text you want displayed for that language. For example:

```
CUSTOM_NamePwdFormToolTip=Forma de Nombre/Clave
```

7d Repeat Step 7c for each supported language file.

8 Restart Tomcat.

```
/etc/init.d/novell-idp restart Or
rcnovell-idp restart
```

Maintaining Customized Identity Server

If you have customized JSP files for Identity Server, you must perform few steps to maintain the customized files before upgrading Access Manager. If you do not, Access Manager overwrites the customized JSP files. For more information, see "Maintaining Customized JSP Files for Identity Server" in the *NetIQ Access Manager Appliance 4.2 Installation and Upgrade Guide*.

Sample Custom Login Pages

- "Modifying the File" on page 215
- "Sample Modified File" on page 219
- "Modified login.jsp File for Credential Prompts" on page 221
- "Custom nidp.jsp File with Custom Credentials" on page 224

NOTE: After customizing a JSP file, you need to sanitize the JSP file to prevent XSS attacks. See, "Preventing Cross-site Scripting Attacks" on page 792.

Modifying the File

The following 4.0 `login.jsp` file has been modified to display line numbers. The lines that require modifications have been highlighted, and a few extra spaces have been added to allow for a better display of the text.

```
1. <%@ page language="java" %>
2. <%@ page pageEncoding="UTF-8" contentType="text/html; charset=UTF-8"%>
3. <%@ page import="com.novell.nidp.common.provider.*" %>
4. <%@ page import="java.util.*" %>
5. <%@ page import="java.net.*" %>
6. <%@ page import="com.novell.nidp.*" %>
7. <%@ page import="com.novell.nidp.servlets.*" %>
8. <%@ page import="com.novell.nidp.resource.*" %>
9. <%@ page import="com.novell.nidp.resource.jsp.*" %>
10.<%@ page import="com.novell.nidp.common.xml.w3c.*" %>
11.<%
12.    response.setHeader("Pragma", "No-cache"); 13.    response.setHeader("Cache-
Control", "no-cache"); 14. 15.    Locale locale = request.getLocale(); 16.
String strLanguageCode = locale.getLanguage(); 17.     String strImageDirectory =
NIDPResourceManager.getInstance().getImage Directory(locale); 18.    NIDPResource
resource = NIDPResourceManager.getInstance().get (JSPResDesc.getInstance(),
locale);
19.%>
20.
21.<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//<%=strLanguage
Code%>">
22.<html lang="<%=strLanguageCode%>">
23.    <head>
24.    <link rel="stylesheet" href="<%= request.getContextPath() %>/images/
hf_style.css" type="text/css">
25.    <style type="text/css" media="screen"><!--
26.      #headimage    { position: relative; top: 0px; left: 0px; z-index: 1}
27.      #title    { position: relative; top: 40px; left: 5px; color: white; z-
index: 4}
28.      #locallabel    { position: relative; top: 78px; left: 10px; z-index: 4}
29.      #login       { text-align: center }
30.         --></style>
31.    <META HTTP-EQUIV="Content-Language" CONTENT="<%=strLanguageCode%>">
32.    <title><%=resource.getString0(JSPResDesc.LOGIN_TITLE)%></title>
33.    <meta http-equiv="content-type" content="text/html;charset=utf-8">
34.    <script type="text/javascript" src="<%= request.getContextPath() %>/images/
showhide_2.js"></script>
35.    <script language="JavaScript">
36.
37.      var i = 0;
38.      function imageSubmit()
39.      {
40.          if (i == 0)
41.          {
42.             i = 1;
43.             document.IDPLogin.submit();
44.          }
45.
46.          return false;
47.      }
48.  </script>
49.    </head>
50.    <body marginwidth="0" marginheight="0" leftmargin="0" topmargin="0"
rightmargin="0" onLoad="document.IDPLogin.Ecom_User_ID.focus();" >
51.        <form name="IDPLogin" enctype="application/x-www-form-urlencoded"
method="POST" action="<%= (String) request.getAttribute("url") %>"
AUTOCOMPLETE="off">
52.        <table style="margin-top: 6em" width="100%" border="0" cellspacing="0"
cellpadding="0">
53.        <tr>
```

```
54.            <td width="50%" height="80 px"> </td>
55.        <td colspan="2">
56.            <div id="title"><b><%=resource.getString0(JSPResDesc.
LOGIN_TITLE)%></b></div>
57.        <div id="locallabel"><b><%=resource.getString0(JSPResDesc.
LOCAL_LOGIN)%></b></div>
58.        <div id="headimage"><img src="<%= request.getContextPath() %>/images/
Odyssey_LoginHead.gif" alt="" height="80" width="550" border="0"></div>
59.                    </td>
60.        <td width="100%"> </td>
61.        </tr>
62.        <tr>
63.            <td width="50%"> </td>
64.        <td style="background-color: #efeee9; padding: 10px" colspan="2">
65.<%
66.    String err = (String) request.getAttribute(NIDPConstants.ATTR _LOGIN_ERROR);
67.    if (err != null)
68.    {
69. %>
70.        <div><label><%=err%></label></div>
71. <%  }
72.
73   // Determine if this login page is being used for account identification 74.
// purposes 75.    String id = (String) request.getAttribute("identify"); 76.    if
(id != null && id.equals("true")) 77.    { 78. %> 79.
<div><%=resource.getString0(JSPResDesc.IDENTIFY)%></div> 80. <%  } %>
81.          <span id="login2" style="display: block;">
82.            <table>
83.              <tr>
84.                <td nowrap="nowrap">
85.                  <div>
86.                    <label style="width: 100px"><%=resource.getString0
(JSPResDesc.USERNAME)%></label></label>
87.                  </div>
88.                </td>
89.                <td width="100%" nowrap="nowrap">
90.                  <div>
91.                    <input type="text" class="smalltext" name="Ecom_User_ID"
size="30">
92.                  </div>
93.                </td>
94.              </tr>
95.              <tr>
96.                <td nowrap="nowrap">
97.                  <div>
98.                   <label><%=resource.getString0(JSPResDesc.PASSWORD)%></label>
99.                  </div>
100.                </td>
101.                <td style="white-space: nowrap">
102.                  <div>
103.                    <input type="password" class="smalltext" name="Ecom_
Password" size="30">  
104.                    <input alt="<%=resource.getString0(JSPResDesc.LOGIN)%>"
border="0" name="loginButton2" src="<%= request.getContextPath() %>/images/
<%=strImageDirectory%>/btnlogin_<%=strImageDirectory%>.gif" type="image"
value="Login" onClick="return imageSubmit()">
105.                  </div>
106.                </td>
107.              </tr>
108.<% 109.  String prov = (String) request.getAttribute("provision"); 110.  if
```

```
(prov != null) 111.  { 112.%> 113.              <tr> 114.                <td
colspan=2> 115.                  <div> 116.                    <label><a
href="<%=prov%>"><%=resource.getString0 (JSPResDesc.CREATE_ACCT)%></a></label>
117.              </div>          118.             </td>            119.
</tr> 120.<%      } %>
121.            </table>
122.          </span>
123.        </td>
124.        <td width="100%"> </td>
125.      </tr>
126. <% 127.  DisplayableProvider[] list = (DisplayableProvider[]) request.get
Attribute("providers"); 128.     if (list != null && list.length > 0) 129.     {
130.%> 131.      <tr> 132.        <td width="50%"></td> 133.        <td
style="background-color: #efeeec; padding-left: 10px; padding-bottom:
10px"colspan="2"> 134.         <div style="margin-left: -10px; background: url(<%=
request.getContextPath() %>/images/dotline_bg.gif) repeat-x"> </div> 135.
<div><b><%=resource.getString0(JSPResDesc.FEDERATED_LOGIN)%></b></div> 136.<% 137.
for (int i = 0; i < list.length; i++) 138.    { 139.%> 140.          <a
style="padding: 5px" href="<%=list[i].getAuthenticationUrl
(request.getContextPath())%>"> 141.<% 142.       if (list[i].hasIcon()) 143.
{          144.%> 145.          <img border=0 class="margin4"
alt="<%=XMLUtil.stringToHTML String(list[i].getDisplayName())%>"
src="<%=XMLUtil.stringToHTMLString (list[i].getIcon(request))%>"
align="absmiddle"></a> 146.<% 147.     } 148.    else 149.    { 150.%>
151.          <%=XMLUtil.stringToHTMLString(list[i].getDisplayName())%></a> 152.<%
153.       } 154.     155.     } %> 156.       </td> 157.       <td width="100%"></
td> 158.      </tr> 159.<%  } %>
160.        <tr>
161.            <td width="50%"></td>
162.        <td style="background-color: #E6D88C; padding-left: 10px"><img
style="padding-right: 200px" src="<%= request.getContextPath() %>/images/
LAP_interoperable_logo_100.gif" align="absmiddle" border="0"></td>
163.        <td style="background-color: #E6D88C; padding-right: 10px"
align="right" width="100">
164.
165.<% 166.  String cancel = (String) request.getAttribute("cancel"); 167.  if
(cancel != null) 168.  { 169.%> 170.                        <input
alt="<%=resource.getString0(JSPResDesc. CANCEL)%>" border="0" name="Cancel"
src="<%= request.getContextPath() %>/images/<%=strImageDirectory%>/
btncancel_<%=strImageDirectory%>.gif" type="image" value="Cancel" tabindex="4">
171.<%    } 172.      else  173.      {  174.%> 175.             176.<%    }
%>
177.        </td>
178.            <td width="100%"></td>
179.        </tr>
180.<%
181.  if (NIDPCripple.isCripple())
182.  {
183.%>
184.        <tr>
185.               <td colspan=4 width="100%" align="center"><%=NIDPCripple.
getCrippleAdvertisement(locale)%></td>
186.        </tr>
187.<%
188.  }
189.%>
190.         </table>
191.        </form>
192.    </body>
193.</html>
```

Sample Modified File

The following file shows all the changes that allow 4.0 `login.jsp` to compile on a 4.0 SP1 Identity Server. The deleted lines have been replaced with returns, so you can line this file up with the original to see the modifications.

```
<%@ page language="java" %>
<%@ page pageEncoding="UTF-8" contentType="text/html; charset=UTF-8"%>
<%@ page import="com.novell.nidp.common.provider.*" %>
<%@ page import="java.util.*" %>
<%@ page import="com.novell.nidp.ui.*" %>
<%@ page import="com.novell.nidp.*" %>
<%@ page import="com.novell.nidp.servlets.*" %>
<%@ page import="com.novell.nidp.resource.*" %>
<%@ page import="com.novell.nidp.resource.jsp.*" %>
<%@ page import="com.novell.nidp.common.xml.w3c.*" %>
<%
ContentHandler handler = new ContentHandler(request,response);

%>

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//
<%=handler.getLanguageCode()%>">
<html lang="<%=handler.getLanguageCode()%>">
    <head>
    <link rel="stylesheet" href="<%= request.getContextPath() %>/images/
hf_style.css" type="text/css">
    <style type="text/css" media="screen"><!--
      #headimage    { position: relative; top: 0px; left: 0px; z-index: 1}
      #title    { position: relative; top: 40px; left: 5px; color: white; z-index:
4}
      #locallabel    { position: relative; top: 78px; left: 10px; z-index: 4}
      #login       { text-align: center }
         --></style>
    <META HTTP-EQUIV="Content-Language" CONTENT="<%=handler.getLanguageCode()%>">
    <title><%=handler.getResource(JSPResDesc.TITLE)%></title>
    <meta http-equiv="content-type" content="text/html;charset=utf-8">
    <script type="text/javascript" src="<%= request.getContextPath() %>/images/
showhide_2.js"></script>
    <script language="JavaScript">

      var i = 0;
      function imageSubmit()
      {
          if (i == 0)
          {
             i = 1;
             document.IDPLogin.submit();
          }

          return false;
      }
  </script>
    </head>
    <body marginwidth="0" marginheight="0" leftmargin="0" topmargin="0"
rightmargin="0" onLoad="document.IDPLogin.Ecom_User_ID.focus();" >
        <form name="IDPLogin" enctype="application/x-www-form-urlencoded"
method="POST" action="<%= (String) request.getAttribute("url") %>"
AUTOCOMPLETE="off">
        <table style="margin-top: 6em" width="100%" border="0" cellspacing="0"
```

```
cellpadding="0">
        <tr>
            <td width="50%" height="80 px"> </td>
        <td colspan="2">
           <div id="title"><b><%=handler.getResource(JSPResDesc.TITLE)%></b></div>
        <div id="locallabel"><b><%=handler.getResource(JSPResDesc.PRODUCT)%></b></
div>
        <div id="headimage"><img src="<%= request.getContextPath() %>/images/
Odyssey_LoginHead.gif" alt="" height="80" width="550" border="0"></div>
                    </td>
        <td width="100%"> </td>
        </tr>
        <tr>
            <td width="50%"> </td>
        <td style="background-color: #efeee9; padding: 10px" colspan="2">
<%
    String err = (String) request.getAttribute(NIDPConstants.ATTR_LOGIN_ERROR);
    if (err != null)
    {
%>
        <div><label><%=err%></label></div>
<%  }



%>
          <span id="login2" style="display: block;">
            <table>
              <tr>
                <td nowrap="nowrap">
                  <div>
                    <label style="width:
100px"><%=handler.getResource(JSPResDesc.USERNAME)%></label></label>
                  </div>
                </td>
                <td width="100%" nowrap="nowrap">
                  <div>
                    <input type="text" class="smalltext" name="Ecom_User_ID"
size="30">
                  </div>
                </td>
              </tr>
              <tr>
                <td nowrap="nowrap">
                  <div>
                    <label><%=handler.getResource(JSPResDesc.PASSWORD)%></label>
                  </div>
                </td>
                <td style="white-space: nowrap">
                  <div>
                    <input type="password" class="smalltext" name="Ecom_Password"
size="30">  
                    <input alt="<%=handler.getResource(JSPResDesc.LOGIN)%>"
border="0" name="loginButton2" src="<%=handler.getImage("btnlogin.gif",true)%>"
type="image" value="Login" onClick="return imageSubmit()">
                  </div>
                </td>
              </tr>
```

```
            </table>
          </span>
        </td>
        <td width="100%"> </td>
      </tr>



        <tr>
            <td width="50%"></td>
        <td style="background-color: #E6D88C; padding-left: 10px"><img
style="padding-right: 200px" src="<%= request.getContextPath() %>/images/
LAP_interoperable_logo_100.gif" align="absmiddle" border="0"></td>
        <td style="background-color: #E6D88C; padding-right: 10px" align="right"
width="100">



        </td>
            <td width="100%"></td>
        </tr>
<%
  if (NIDPCripple.isCripple())
  {
%>
        <tr>
               <td colspan=4 width="100%"
align="center"><%=NIDPCripple.getCrippleAdvertisement(request.getLocale())%></td>
        </tr>
<%
  }
%>
        </table>
        </form>
    </body>
</html>
```

Modified login.jsp File for Credential Prompts

The following code is a modified version of the 3.2 `login.jsp` file. It has been modified to add a prompt for the user's email address.

Such a JSP file must be used with a contract that uses a method that defines the query for the new attribute. The method also needs to define which login file has been modified to display the prompt. For more information about this process, see "Customizing the Default Login Page to Prompt for Different Credentials" on page 195.

The sample code contains the following the text for the prompt:

```
<td align=left>
   <label>Email Address:</label>
</td>
```

It also adds an input element for the query variable:

```
<td align=left>
   <input type="text" class="smalltext" name="Ecom_User_Mail" size="30">
</td>
```

These elements are both part of the new <tr> element that has been added to the file. These lines are marked in bold in the following sample file.

```
<%@ page language="java" %>
<%@ page pageEncoding="UTF-8" contentType="text/html; charset=UTF-8"%>
<%@ page import="java.util.*" %>
<%@ page import="com.novell.nidp.*" %>
<%@ page import="com.novell.nidp.servlets.*" %>
<%@ page import="com.novell.nidp.resource.*" %>
<%@ page import="com.novell.nidp.resource.jsp.*" %>
<%@ page import="com.novell.nidp.ui.*" %>
<%
    ContentHandler handler = new ContentHandler(request,response);
    String target = (String) request.getAttribute("target");
%>

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//
<%=handler.getLanguageCode()%>">
<html lang="<%=handler.getLanguageCode()%>">
  <head>
        <META HTTP-EQUIV="Content-Language"
CONTENT="<%=handler.getLanguageCode()%>">
      <meta http-equiv="content-type" content="text/html;charset=utf-8">

    <style type="text/css" media="screen">
      td label       { font-size: 0.85em ; padding-right: 0.2em; }
      label     { font-size: 0.77em; padding-right: 0.2em; }
         input { font-family: sans-serif; }
      .instructions   { color: #4d6d8b; font-size: 0.8em; margin: 0 10px 10px 0 }
    </style>

       <script type="text/javascript" src="<%=
handler.getImage("showhide_2.js",false)%>"></script>
      <script language="JavaScript">
      var i = 0;
      function imageSubmit()
      {
            if (i == 0)
            {
              i = 1;
              document.IDPLogin.submit();
            }

            return false;
      }
    </script>
      </head>
  <body style="background-color: <%=handler.getBGColor()%>" marginwidth="0"
marginheight="0" leftmargin="0" topmargin="0" rightmargin="0"
onLoad="document.IDPLogin.Ecom_User_ID.focus();" >
        <form name="IDPLogin" enctype="application/x-www-form-urlencoded"
method="POST" action="<%= (String) request.getAttribute("url") %>"
AUTOCOMPLETE="off">
      <input type="hidden" name="option" value="credential">
<% if (target != null) { %>
      <input type="hidden" name="target" value="<%=target%>">
```

```
<% } %>
          <table border=0 style="margin-top: 1em" width="100%" cellspacing="0"
cellpadding="0">
            <tr>
            <td style="padding: 0px">
            <table border=0>
              <tr>
                <td align=left>
                  <label><%=handler.getResource(JSPResDesc.USERNAME)%></label>
                </td>
                <td align=left>
                  <input type="text" class="smalltext" name="Ecom_User_ID"
size="30">
                </td>
              </tr>
              <tr>
                <td align=left>
                  <label>Email Address:</label>
                </td>
                <td align=left>
                  <input type="text" class="smalltext" name="Ecom_User_Mail"
size="30">
                </td>
              </tr>
               <tr>
                <td align=left>
               <label><%=handler.getResource(JSPResDesc.PASSWORD)%></label>
                </td>
                <td align=left>
                  <input type="password" class="smalltext" name="Ecom_Password"
size="30">
                </td>
              </tr>
              <tr>
                <td align=right colspan=2 style="white-space: nowrap">
                <input alt="<%=handler.getResource(JSPResDesc.LOGIN)%>" border="0"
name="loginButton2" src="<%= handler.getImage("btnlogin.gif",true)%>" type="image"
value="Login" onClick="return imageSubmit()">
                </td>
              </tr>
            </table>
          </td>
        </tr>
<%
    String err = (String) request.getAttribute(NIDPConstants.ATTR_LOGIN_ERROR);
    if (err != null)
    {
%>
            <td style="padding: 10px">
            <div class="instructions"><%=err%></div>
            </td>
                   </tr>
<%  } %>
<%
```

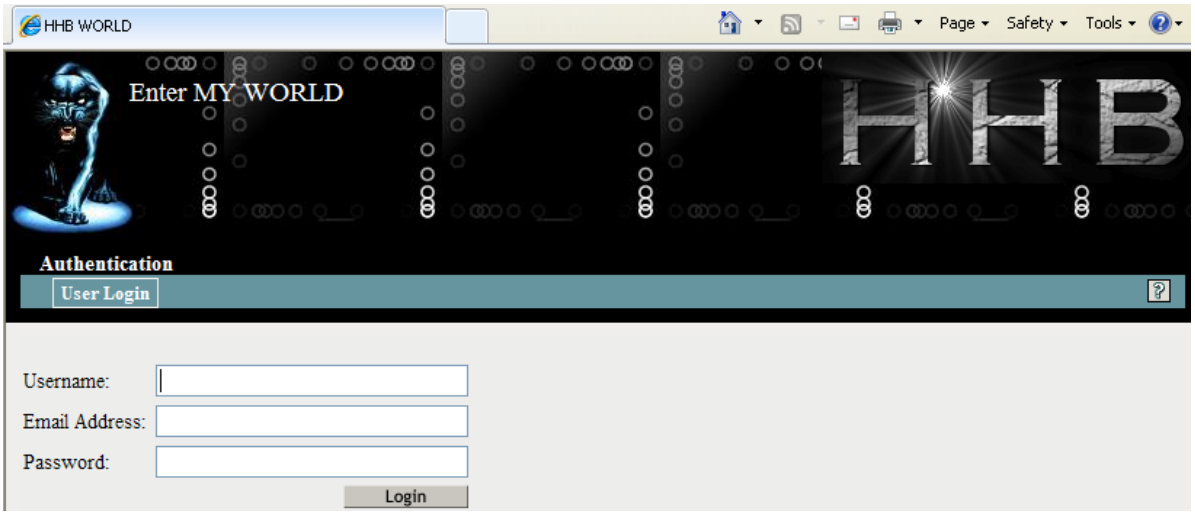
```
  if (NIDPCripple.isCripple())
  {
%>
            <tr>
                   <td width="100%"
align="center"><%=NIDPCripple.getCrippleAdvertisement(request.getLocale())%></td>
            </tr>
<%
  }
%>
          </table>
          </form>
      </body>
</html>
```

Custom nidp.jsp File with Custom Credentials

To create a custom `nidp_legacy.jsp` file that uses custom credentials, you need to modify the `nidp_legacy.jsp` file, create a method and contract for the file, and modify the `main.jsp` file. For instructions, see “Customizing the nidp.jsp File” on page 197 and “Adding Logic to the main.jsp File” on page 204.

Figure 4-3 illustrates the login page that the following custom `nidp_legacy.jsp` file and `main.jsp` file create.

Figure 4-3 *Custom Branding with Custom Credential Prompts*



The credential frame uses the same modifications in the sample from “Modified login.jsp File for Credential Prompts” on page 221. The following sections provide the other required sample files to create this login page and information about the required method and contract:

- “The Modified nidp.jsp File” on page 224
- “The Modified main.jsp File” on page 229
- “The Method and the Contract” on page 230

The Modified nidp.jsp File

The background, menu, and border colors are set to black. These colors are specified in the following lines in the sample file:

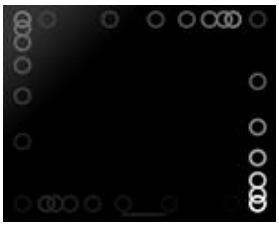
```
// Background color
String bgcolor   = "#000000";

// Menu color
String menucolor   = "#000000";

// Border color
String bcolor   = "#000000";
```

Figure 4-4 illustrates the image (`images2.jpeg`) that this custom page uses for the header background image:

Figure 4-4 *Background Image*



This image is the repeatable image that allows the header to be resized. This image is specified in the following lines in the file:

```
// The header background image that gets repeated
   String hdrBgndImg  = "/custom_images/images2.jpeg";
```

Figure 4-5 illustrates the image (`images3.jpeg`) that this custom page uses for the product logo that appears on left of the header frame.

Figure 4-5 *Header Image*



Figure 4-6 illustrates the image (`hhbimages.jpeg`) that this custom page uses to replace the Novell company logo on the right of the header frame.

Figure 4-6 *Company Logo*



The following lines define what appears as the title for the browser window:

```
<title>HHB WORLD</title>
```

The following line defines the header title value:

```
String hdrTitle     = "Enter MY WORLD";
```

Its position is controlled by the following line in the file:

```
 #title { position: absolute; font-size: 1.2em; color: white; top: 18px; left:
85px; }
```

The top position has been modified from 13px to 18px and the left position has been modified from 55px to 85px. The other lines in this section control the position of the other items in the header.

The lines that have been modified are marked in bold in the following file.

```
<%
    ContentHandler handler = new ContentHandler(request,response);

    // Background color
    String bgcolor   = "#000000";

    // Menu color
    String menucolor   = "#000000";

    // Border color
    String bcolor   = "#000000";

    // The header background image that gets repeated
    String hdrBgndImg  = "/custom_images/images2.jpeg";

    String hdrImage  = "/custom_images/images3.jpeg";

    String hdrLogo      = "/custom_images/hhbimages.jpeg";

    String hdrTitle     = "Enter MY WORLD";

    String query     =  request.getQueryString();
    if (query != null && query.length() > 0)
        query = "&" + query;
    else query = "";
%>

<!DOCTYPE HTML PUBLIC "-//W3C//Dtd HTML 4.0 transitional//
<%=handler.getLanguageCode()%>">
<html lang="<%=handler.getLanguageCode()%>">
    <head>
        <title>HHB WORLD</title>
        <meta http-equiv="content-type" content="text/html; charset=UTF-8">
        <link href="<%= handler.getImage("hf_menu.css",false)%>" rel="stylesheet">
        <link href="<%= handler.getImage("HF_message.css",false)%>"
rel="stylesheet">
        <link href="<%= handler.getImage("HF_obj_list_table.css",false)%>"
rel="stylesheet">
    <style>
    * { margin: 0; padding: 0; }
        #header    { background-image: url(<%=
handler.getImage(hdrBgndImg,false)%>); background-repeat: repeat-x; }
        #logo      { position: absolute; top: 0px; right: 0px;  }
        #title         { position: absolute; font-size: 1.2em; color: white; top:
18px; left: 85px; }
    #subtitle    { position: relative; font-size: .9em; color: black; white-space:
nowrap; top: 0px; left: 0px; text-align: right; }
    #mcontent     { position: relative; padding: 5px; background-color:
<%=bgcolor%>; }
    #content      { width: 100%; border: 0; margin: 0; padding: 0; overflow: none;
height: 376px; background-color: <%=bgcolor%>;}
    #logoutbut    { position: absolute; top: 25px; right: 35px;  }
    #helpbutlogin  { position: absolute; color: yellow; top: 25px; right: 10px; }
```

```
    #loggingbut    { position: absolute; color: blue; top: 25px; right: 65px;  }

    .NLtab .tab1s    { background-color: <%=menucolor%>; padding-left: 3px;
padding-right: 8px; text-align: center; white-space: nowrap;   }
    .NLtab .tab1s a   { text-decoration: none; }
    .NLtab span.tab1s { padding:5; color: white; font-size: 0.9em; font-weight:
bold; line-height: 17px; background-color: transparent; background-image: none;
text-decoration: none; }
    .NLtab .tab1u   { background-color: <%=bgcolor%>; padding-left: 3px; padding-
right: 3px; text-align: center; white-space: nowrap; border-left: 1px solid
<%=bcolor%>; border-right: 1px solid <%=bcolor%>; border-top: 1px solid
<%=bcolor%>;          }
    .NLtab span.tab1u { border: none; padding:5; color: black; font-size: 0.8em;
font-weight: bold; line-height: 17px; text-decoration: none; background-color:
transparent;  }

    .NLtab tr.subtab td  {  color: white; padding: 2px }
    .NLtab tr.subtab a  { font-size: .8em; color: white; text-decoration: none;
padding: 2px 5px 2px 5px}

    .selx  { border: 1px solid rgb(239, 238, 236); font-size: 1em; font-weight:
bolder; background-repeat: repeat-x; background-position: 0pt bottom;}
   .unselx  { border: 0px; font-size: .9em; font-weight: normal; background-image:
none; }
        </style>

        <script>
         var g_curCard = null;     // initial displayed card
     var g_cardContainer = null;   // div that holds all the authentication cards
     var g_curSubtab = null;   // subtab currently displayed
     var g_curTab = null;     // tab currently displayed

         var menuItem = 0;
             function showHide(i)
             {
                 document.getElementById('menu1').style.display='none';
                 document.getElementById('menu2').style.display='none';
                 document.getElementById('submenu1').style.display='none';
                 document.getElementById('submenu2').style.display='none';
                 document.getElementById('menu' + i).style.display='block';
                 document.getElementById('submenu' + i).style.display='block';
         if (i == 1)
             switchContentPage("<%= handler.getJSP("content")%>");
         else
             switchContentPage("<%= handler.getJSP("IdentityEditor")%>");
             }

     function switchContentPage(newSrc)
     {
           parent.document.getElementById("content").src = newSrc;
     }

     function onloadhandler()
     {
         g_cardContainer = document.getElementById("cardcontainer");
         g_curSubtab     = document.getElementById("loginsubtab");
         g_curTab        = document.getElementById("authtab");
         g_curCard       = document.getElementById("selectedCard0");
     }
```

```
 function showhideTab(divid)
{
    var element1 = document.getElementById(divid);

    if(element1.style.display == "none")
    {
        element1.style.display = "block";
    g_curTab.style.display = "none";

        g_curTab = element1;
    }
}

function subtabchange(divid)
 {
    var element1 = document.getElementById(divid);
    var element2 = g_curSubtab;
    element1.className = "selx";
    if (element1.id != element2.id)
    {
        element2.className = "unselx";
    }
    g_curSubtab = element1;
}

function showHelp()
 {
    var helpURL = "login.html";
            if (g_curSubtab.id == "fedsubtab")
        helpURL = "<%=handler.getHelp("federations.html")%>";

            else if (g_curSubtab.id == "myprofile")
    helpURL = "<%=handler.getHelp("myprofile.html")%>";

            else if (g_curSubtab.id == "sharing")
     helpURL = "<%=handler.getHelp("sharing.html")%>";

            else if (g_curSubtab.id == "loginsubtab")
    helpURL = "<%=handler.getHelp("userlogin.html")%>";

            else if (g_curSubtab.id == "newcardsubtab")
    helpURL = "<%=handler.getHelp("newcard.html")%>";

          else if (g_curSubtab.id == "logTicketsubtab")
        helpURL = "<%=handler.getHelp("logticket.html")%>";

        var w;
        w = window.open(helpURL, "nidsPopupHelp",
"toolbar=no,location=no,directories=no,menubar=no,scrollbars=yes,resizable=yes,wid
th=500,height=500");
        if (w != null)
        {
         w.focus();
        }
}
   </script>
</head>

<body onload="onloadhandler()">
   <table width=100% border=0 cellpadding=0 cellspacing=0 bgcolor=<%=bgcolor%>
```

```
>
            <tr>
        <td>
            <table cellspacing=0 width=100% border=0>
            <tr>
            <td width=100%>
               <div id="header"><img src="<%=handler.getImage(hdrImage,false)%>"></
div>
                <div id="logo"><img src="<%=handler.getImage(hdrLogo,false)%>"></
div>
                <div id="title"><%=hdrTitle%></div>
            </td>
        </tr>
        </table>
        </td>
    </tr>
    <tr>
        <td>
            <table cellspacing=5 width=100%>
            <tr>
            <td>
                <%@ include file="menus.jsp" %>
            </td>
        </tr>
        </table>
        </td>
    </tr>
    <tr>
        <td>
            <table cellspacing=0 border=0 width=100%>
            <tr>
            <td>
                          <iframe scrolling=no id="content"
src="<%=handler.addCardParm(handler.getJSP(handler.isJSPMsg() ?
handler.getJSPMessage().getJSP() : NIDPConstants.JSP_CONTENT)) + query%>"
frameborder=0></iframe>
            </td>
        </tr>
        </table>
        </td>
    </tr>
    </table>
    </body>
</html>
```

The Modified main.jsp File

The following sample file has two types of modifications. The following line has been added so that the URI of the contract can be read and used as a condition for selecting the login page to display:

```
String strContractURI = hand.getContractURI();
```

The following lines define the login page to use when the URI of the contract is set to login/custom.

```
else if(strContractURI != null && strContractURI.equals("login/custom"))
    {
%>
     <%@ include file="custom.jsp" %>

<%  }
```

The lines that have been added are marked in bold in the following file.

```
<%@ page language="java" %>
<%@ page pageEncoding="UTF-8" contentType="text/html; charset=UTF-8"%>
<%@ page import="com.novell.nidp.*" %>
<%@ page import="com.novell.nidp.resource.jsp.*" %>
<%@ page import="com.novell.nidp.ui.*" %>
<%@ page import="com.novell.nidp.common.util.*" %>
<%@ page import="com.novell.nidp.liberty.wsf.idsis.apservice.schema.*" %>

<%
    ContentHandler hand = new ContentHandler(request,response);
    String strContractURI = hand.getContractURI();

    // Is there a JSP defined on a class definition or a method definition
    // that must be displayed as the main jsp here?
    if (hand.contractDefinesMainJSP())
    {
%>

        <%@ include file="mainRedirect.jsp" %>
<%  }

else if(strContractURI != null && strContractURI.equals("login/custom"))
    {
%>
     <%@ include file="custom.jsp" %>

<%  }

    // This is the jsp used by default
    else
    {
%>
        <%@ include file="nidp.jsp" %>
<%  }  %>
```

The Method and the Contract

After modifying the two files, you still need to create a method and a contract. The method needs to use a name/password class and have the following properties defined:

- Query property values:

  Property Name: `Query`

  Property Value: `(&(objectclass=person)(mail=%Ecom_User_Mail%))`

- JSP property values:

  Property Name: `JSP`

  Property Value: <*filename*>

  Replace <*filename*> with the name of your login page that modifies the credential prompts. Do not include the JSP extension in the value.

You then need to create a contract that uses this method and assign it to a protected resource.

Access Gateway Server Advance Configuration

This section describes the configuration settings that affect the Access Gateway as a server, such as changing its name or setting the time.

- “Configuration Overview” on page 231
- “Saving, Applying, or Canceling Configuration Changes” on page 232
- “Managing Access Gateways” on page 234
- “Managing General Details of the Access Gateway” on page 238
- “Setting Up a Tunnel” on page 240
- “Setting the Date and Time” on page 241
- “Configuring Network Settings” on page 242
- “Enabling the Access Gateway to Display Post-Authentication Message” on page 246
- “Customizing The Access Gateway” on page 246

For logging and audit options, see the following:

- “Managing Access Gateway Logs” on page 860
- “Configuring Logging for a Proxy Service” on page 861
- “Enabling Access Gateway Audit Events” on page 840
- “Access Gateway Advanced Options” on page 259

Configuration Overview

The Configuration page allows you to view the configuration status and to configure the features of the cluster or the Access Gateway. After an Access Gateway has been made a member of a cluster, you can only configure it from the cluster configuration. Some options are specific to an Access Gateway. For these options, you must select the Access Gateway and then configure the options.

1 In the Administration Console, **Devices** > **Access Gateways** > **Edit**.

To edit an Access Gateway that is not a member of a cluster, click the **Edit** button on the Access Gateway row.

To edit an Access Gateway cluster, click the **Edit** button on the Access Gateway cluster row.

2 Select one of the following options:

Reverse Proxy / Authentication: Allows you to configure a reverse proxy so that it hides the IP address of a Web server and accelerates access by caching the most frequently used pages. This option displays the list of configured proxies and allows you to add new proxies and modify existing proxies. To add a new reverse proxy or manage the existing proxies, click **Reverse Proxy / Authentication** (see “Managing Reverse Proxies and Authentication” on page 94). To manage a specific reverse proxy, click its name (see “Creating a Proxy Service” on page 95).

Tunneling: Allows you to tunnel non-HTTP traffic through the Access Gateway to a Web server. For more information, see “Setting Up a Tunnel” on page 240.

Date & Time: Allows you to configure the server’s time source. For more information, see “Setting the Date and Time” on page 241.

Alerts: Allows you to select the alerts and then configure whether they are sent to a server, a log file, or to selected individuals via e-mail. For more information, see “Managing Access Gateway Alert Profiles” on page 912.

Auditing: Allows you to select the events to send to a NetIQ Sentinel or Audit server. For more information, see “Enabling Access Gateway Audit Events” on page 840.

Adapter List: Displays the list of configured network cards and allows you to edit an existing configuration or to add a new one. For more information, see “Viewing and Modifying Adapter Settings” on page 242. To manage a specific adapter, click the name of the adapter.

Gateways: Displays the list of configured gateways and allows you to edit an existing configuration or to add a new one. For more information, see “Viewing and Modifying Gateway Settings” on page 243.

DNS: Displays the current DNS configuration that the Access Gateway is using to resolve names and allows you to modify it. For more information, see “Viewing and Modifying DNS Settings” on page 244.

Hosts: Allows you to create a static mapping between the host IP addresses and host names. For more information, see “Configuring Hosts” on page 245.

Purge List: Allows you to prevent Web objects from being cached. For more information, see “Configuring a Purge List” on page 257.

Pin List: Allows you to prepopulate the cache with the Web objects that you want cached, before a user has requested the object. For more information, see “Configuring a Pin List” on page 254.

Cache Options: Allows you to globally disable caching or configure which objects are cached and how frequently they are refreshed. For more information, see Configuring Caching Options.

Advanced Options: Allows you to configure how all reverse proxies handle specific items in cache. For more information, see “Configuring the Global Advanced Options” on page 259.

3 For information about using the **OK**, **Cancel**, and **Revert** buttons, see “Saving, Applying, or Canceling Configuration Changes” on page 232.

Saving, Applying, or Canceling Configuration Changes

When you make configuration changes on a page accessed from **Devices** > **Access Gateways** > **Edit** and click **OK** on that page, the changes are saved to the browser cache. If your session expires or you close the browser session before you update the Access Gateway with the changes, the changes are lost.

The Configuration page allows you to control how your changes are saved so they can be applied with the update options (see “Configuration Options” on page 236).

If you have any configuration changes saved to the browser cache, use the following options to control what happens to the changes:

OK: To save the configuration changes to the configuration store, click **OK**. This allows you to return at a later time to review or modify the changes before they are applied. If your Access Gateways are clustered and you prefer to update them one at a time, you need to save the configuration change. This ensures that the changes aren’t lost before the last cluster member is updated. When your session times out or you log out, the configuration changes are flushed from the browser cache. If this happens before the changes have been applied to some servers in the cluster, the changes cannot be applied to those servers.

If you decide to cancel the saved changes, click the **Revert** button and the saved configuration is overwritten by the last successfully applied configuration.

Cancel: To cancel changes that are pending in the browser cache, click the **Cancel** button. To cancel modifications to specific services, click the **Cancel** link by the service. The **Cancel** button does not affect the changes that have been saved to the configuration store.

Revert: To cancel any saved changes, click **Revert**, then confirm the cancellation. The saved configuration is overwritten by the last successfully applied configuration.

If you have applied the changes to one member of the cluster, you cannot use the **Revert** button to revert to the configuration you had before applying the changes. If you decide you do not want to apply these changes to other members of the cluster, remove the server that you updated with the changes from the cluster. Then click **Revert** to cancel the saved changes. The members of the cluster return to the last successfully applied configuration. To apply this configuration to the removed server, add this server to the cluster.

The **Revert** button and the **Cancel** button cannot cancel the following configuration changes:

- **Identity Server Cluster:** If you change the **Identity Server Cluster** option on the Reverse Proxy/Authentication page, then click **OK**, the **Revert** button cannot cancel this change. It is saved, and the next time you apply a configuration change, the Identity Server cluster configuration is applied. To cancel the change, you need to return to the Reverse Proxy/Authentication page, set the **Identity Server Cluster** option to the original selection, then click **OK on the Configuration page**.
- **Reverse Proxy for the Embedded Service Provider:** If you change the **Reverse Proxy** option on the Reverse Proxy/Authentication page, then click **OK**, the **Revert** button cannot cancel this change. It is saved, and the next time you apply a configuration change, the **Reverse Proxy** option change is applied. To cancel the change, return to the Reverse Proxy/Authentication page, set the **Reverse Proxy** option to the original selection, then click **OK** on the Configuration page.
- **Port of the Reverse Proxy for the Embedded Service Provider:** If you change the port of the reverse proxy that is used by the Embedded Service Provider (click **Edit** > **[Name of Reverse Proxy]**), then click **OK**, the **Revert** button cannot cancel this change. It is saved, and the next time you apply a configuration change, the port change is applied. To cancel the change, return to the Reverse Proxy page, set the port to the original value, then click **OK** on the Configuration page.
- **Published DNS Name of the Proxy Service for the Embedded Service Provider:** If you change the Published DNS Name of the proxy service that is used by the Embedded Service Provider (click **Edit** > **[Name of Reverse Proxy]** > **[Name of Proxy Service]**), then click **OK**, the **Revert** button cannot cancel this change. It is saved, and the next time you apply a configuration change, the Published DNS Name is changed. To undo the change, return to the Proxy Service page, set the Published DNS Name to its original value, then click **OK** on the Configuration page.
- **Certificates:** Certificates are pushed as soon as they are selected. If you change the server certificate for the reverse proxy (click **Edit** > **[Name of Reverse Proxy]**) or change the Web server certificates (click **Edit** > **[Name of Reverse Proxy]** > **[Name of Proxy Service]** > **Web Servers)**, the **Revert** button cannot cancel these changes. To undo the change, return to the page, select the original certificate, then click **OK**.
- **Renaming a Reverse Proxy:** If you change the name of a reverse proxy (click **Edit** > **Reverse Proxies / Authentication**), then click **OK**, you cannot cancel this change. To undo the change, return to the Reverse Proxies / Authentication page, rename the reverse proxy to its original name, then click **OK** and update the Access Gateway.

Managing Access Gateways

The following sections contain information about settings available with Access Gateways, changing the settings, and their impact on users:

- “Viewing and Modifying Gateway Settings” on page 234
- “Configuration Options” on page 236
- “Scheduling a Command” on page 238

Viewing and Modifying Gateway Settings

Use the Servers page to view the status of Access Gateways, to modify their configuration, and to perform other actions such as creating a new cluster or stopping and starting an Access Gateway or its Embedded Service Provider.

1. In the Administration Console, click **Devices** > **Access Gateways**.
2. Select one of the following:

   Stop: To stop an Access Gateway, select the service, then click **Stop**. You can use the **Restart** option to start the Access Gateway.

   Restart: To stop and start an Access Gateway, select it, then click **Restart**. If the Access Gateway is already stopped, use **Restart** to start it.

   Refresh: To update the list of Access Gateways and the status columns (**Status**, **Health**, **Alerts**, **Commands**), click **Refresh**.
3. To perform an action available in the **Actions** drop-down menu, select an Access Gateway, then select one of the following:

   Schedule Restart: To schedule when the selected Access Gateway must be stopped and then started, select **Schedule Restart**. On an Access Gateway Service, a restart stops the Access Gateway Service, then starts it. For information about how to schedule this command, see “Scheduling a Command” on page 238.

   Schedule Stop: To schedule when the selected Access Gateway or cluster must be stopped, select **Schedule Stop**.

   You can use the **Restart** option to start it again.

   For more information about how to schedule this command, see “Scheduling a Command” on page 238

   Purge List Now: Click **Purge List Now** to cause all objects in the current purge list to be purged from the cache of the selected server or cluster.

   Purge All Cache: Click **Purge All Cache** to purge the server cache for the selected server or cluster. All cached content is lost.

   When you make certain configuration changes such as updating or changing certificates, changing the IP addresses of Web servers, or modifying the rewriter configuration, you are prompted to purge the cache. The cached objects must be updated for users to see the effects of such configuration changes. If your Access Gateways are in a cluster, you need to manage the purge process so your site remains accessible to your users. You must apply the configuration changes to one member of a cluster. When its status returns to healthy and current, issue the command to purge its cache. Then apply the changes to the next cluster member.

   IMPORTANT: Do not issue a purge cache command when an Access Gateway has a pending configuration change. Wait until the configuration change is complete.

Update Health from Server: Click this action to send a request to the server for updated health information. If you have selected multiple servers, a request is sent to each one. The health status changes to an animated circle until the reply returns.

Service Provider: Select one of the following actions:

- **Start Service Provider:** To start the Embedded Service Provider associated with the selected Access Gateway, click **Start Service Provider**. The Embedded Service Provider is the module within the Access Gateway that communicates with the Identity Server.

  The service provider must be restarted whenever you enable or modify logging on the Identity Server.

- **Stop Service Provider:** To stop the Embedded Service Provider associated with the selected Access Gateway, click **Stop Service Provider**. The Embedded Service Provider is the module within the Access Gateway that communicates with the Identity Server.

  When an Access Gateway is not functioning correctly, you must always try stopping and starting the service provider before stopping and starting the Access Gateway.

- **Restart Service Provider:** To restart the Embedded Service Provider associated with the selected Access Gateway, click **Restart Service Provider**. This command stops the Embedded Service Provider and then starts it. The Embedded Service Provider is the module within the Access Gateway that communicates with the Identity Server.

  When an Access Gateway is not functioning correctly, you must always try restarting the service provider before stopping and starting the Access Gateway.

4 Use the following links to manage a cluster or an Access Gateway.

Name: Displays a list of the Access Gateway servers and the clusters that can be managed from this Administration Console.

- To view or modify the general details of a particular server, click the name of the server.
- To view or modify general details of a cluster, click the name of the cluster.

Status: Indicates the configuration status of the clusters and the Access Gateways. Possible states are pending, update, current, and update all. For more information, see “Configuration Options” on page 236.

Health: Indicates whether a cluster or an Access Gateway is functional. Click the icon to view additional information about the operational status of an Access Gateway.

- For information about the health of a specific Access Gateway, click the health icon on the Access Gateway row. For more information, see “Monitoring the Health of an Access Gateway” on page 904.
- For information about the health of a Access Gateway cluster, click the health icon on the cluster row. For more information, see “Monitoring the Health of an Access Gateway Cluster” on page 906.

Alerts: Indicates whether any alerts have been sent. If the alert count is non-zero, click the count to view more information.

- For information about the alerts of a specific Access Gateway, click the link on the Access Gateway row. For more information, see “Viewing Access Gateway Alerts” on page 911.
- For information about the alerts sent to the cluster, click the link on the cluster row. For more information, see “Viewing Access Gateway Cluster Alerts” on page 912.

Commands: Indicates the status of the last executed command and whether any commands are pending. Click the link to view more information. For more information, see “Viewing the Command Status of the Access Gateway” on page 908.

Statistics: Provides a link to the statistic pages.

- For information about the statistics of a specific Access Gateway, click the **View** link on the Access Gateway row and see "Monitoring Access Gateway Statistics" on page 882.
- For information about statistics sent to the cluster, click the **View** link on the cluster row and see "Monitoring Access Gateway Cluster Statistics" on page 891.

Edit: Provides a link to the configuration page. If the server belongs to a cluster, the **Edit** link appears on the cluster row. Otherwise, the link is on the server row. See "Configuration Overview" on page 231.

Configuration Options

Use the information in this section to modify the Status options described in Step 4 on page 235.

1. In the Administration Console, click **Devices** > **Access Gateways**.
2. View the **Status** column and make changes as necessary.

Status	Description
Current	Indicates that all configuration changes have been applied.
Update	Indicates that a configuration change has been made, but not applied. To apply the changes, click the Update link, then select one of the following:. ◆ **All Configuration:** The **All Configuration** option causes the Access Gateway to read its complete configuration file and restarts the Embedded Service Provider. The configuration update causes logged-in users to lose their connections unless the server is a member of a cluster. When the server is a member of a cluster, the users are sent to another Access Gateway and they experience no interruption of service. ◆ **Logging Settings:** When the ESP logging settings have been modified on the Identity Server, the update option for **Logging Settings** is available. The **Logging Settings** option causes no interruption in services. When you modify Access Gateway logging settings, this option is not available because they are considered configuration settings. ◆ **Policy Settings:** If a policy is modified for a protected resource of the Access Gateway and the policy change is the only modification that has occurred, the update option for **Policy Settings** is available. This option causes no interruption in services. ◆ **Rewriter Profile Changes:** When the administrator changes the rewriter profile, a purge cache command is issued to a Gateway from the administration console, the connection is lost and the service is interrupted for a few seconds. Similar experience is observed during the rewriter profile configuration change, as this internally triggers the purge cache command. ◆ **Changing Certificates:** When a certificate configuration is changed from the administration console, the service is interrupted due to the Tomcat restart.

Status	Description
Update All	This link is available when a server belongs to a cluster. You can select to update all the servers at the same time, or you can select to update them one at a time. If the modification is a policy or a logging change, then use **Update All**. If the modification is a configuration change, we recommend that you update the servers one at a time. ◆ When you select **Update All** for a configuration change, users experience an interruption of service. ◆ When you update servers one at a time for a configuration change, users experience no interruption of service. When you make the following configuration changes, the **Update All** option is the only option available and your site will be unavailable while the update occurs: ◆ The Identity Server configuration that is used for authentication is changed (**Access Gateways** > **Edit** > **Reverse Proxy/Authentication**, then select a different value for the **Identity Server Cluster** option). ◆ A different reverse proxy is selected to be used for authentication (**Access Gateways** > **Edit** > **Reverse Proxy/Authentication**, then select a different value for the **Reverse Proxy** option). ◆ The protocol or port of the authenticating reverse proxy is modified (**Access Gateways** > **Edit** > **Reverse Proxy/Authentication** > **[Name of Reverse Proxy]**, then change the SSL options or the port options). ◆ The published DNS name of the authentication proxy service is modified (**Access Gateways** > **Edit** > **Reverse Proxy/Authentication** > **[Name of Reverse Proxy]** > **[Name of First Proxy Service]**, then modify the **Published DNS Name** option). For more information, see "Applying Changes to the Access Gateway Cluster Members" on page 91.
Update !	If the configuration update contains a configuration error, the **Update** link is disabled and the **Configuration Error** icon is displayed. Click the icon to discover which objects have been misconfigured. You need to fix the error by either canceling or modifying the changes before you can perform an update.
Update All !	If the configuration update contains a configuration error, the **Update All** and the member **Update** links are disabled and the **Configuration Error** icon is displayed. Click the icon to discover which objects have been misconfigured. You need to fix the error by either canceling or modifying the changes before you can perform an update.
Pending	Indicates that the server is processing a configuration change, but has not completed the process.
Locked	Indicates that another administrator is making configuration changes. Before you proceed with any configuration changes, you need to coordinate with this administrator and wait until the Access Gateway has been updated with the other administrator's changes.

Scheduling a Command

Use the Schedule New Command page to schedule a command, such as a shutdown, restart, or upgrade.

1. In the Administration Console, click **Devices** > **Access Gateways**.
2. (Conditional) To schedule a shutdown or restart, select a server, then click **Actions** > **Schedule Restart** or **Schedule Stop**. Continue with Step 3.
3. Fill in the following fields:

   Name Scheduled Command: (Required) Specify a name for this scheduled command. This name is used in log files.

   Description: (Optional) Specify a reason for the command.

   Date & Time: Select the day, month, year, hour, and minute when the command must execute.

   The following fields display information about the command you are scheduling:

   Type: Displays the type of command that is being scheduled, such as Access Gateway Shutdown, Access Gateway Restart, or Access Gateway Upgrade.

   Server: Displays the name of the server that the command is being scheduled for.
4. Click **OK** to schedule the command.

Managing General Details of the Access Gateway

The Server Details page allows you to perform general maintenance actions on the selected Access Gateway.

1. In the Administration Console, click **Devices** > **Access Gateways** > **[Name of Access Gateway]**.
2. Select one of the following options:

   Edit: Click this option to edit the general details of the Access Gateway. See “Changing the Name of an Access Gateway and Modifying Other Server Details” on page 238.

   New IP: Click this action to trigger a scan to detect new IP addresses. This might take some time. If you have used a system utility to add an IP address after you have installed the Access Gateway Service, use this option to update the Access Gateway Service to display the new IP address as a configuration option. For more information about this option, see “Adding a New IP Address to the Access Gateway” on page 245.

   Configuration: Click this option to export the configuration of this Access Gateway or to import the configuration of a saved configuration file. See “Exporting and Importing an Access Gateway Configuration” on page 239.
3. Click **Close**.

Changing the Name of an Access Gateway and Modifying Other Server Details

The default name of an Access Gateway is its IP address. You can change this to a more descriptive name as well as modifying other details that can help you identify one Access Gateway from another.

1. In the Administration Console, click **Devices** > **Access Gateways** > **[Name of Access Gateway]** > **Edit**.
2. Modify the values in the following fields:

Name: Specify the Administration Console display name for the Access Gateway. This is a required field. The default name is the IP address of the Access Gateway. If you modify the name, the name must use alphanumeric characters and can include spaces, hyphens, and underscores.

Management IP Address: Specify the IP address used to manage the Access Gateway. Select an IP address from the list.

Port: Specify the port to use for communication with the Administration Console.

Location: Specify the location of the Access Gateway server. This is optional, but useful if your network has multiple Access Gateway servers.

Description: Describe the purpose of this Access Gateway. This is optional, but useful if your network has multiple Access Gateways.

3 Click **OK** twice, then click **Close**.

When you click **OK**, any changes are immediately applied to the Access Gateway.

Exporting and Importing an Access Gateway Configuration

You can export an existing Access Gateway configuration and its dependent policies, and then import this configuration to a new server. This feature is especially useful for deployments that set up configurations in a staging environment, test and validate the configuration, then want to deploy the configuration on new hardware that exists in the production environment.

IMPORTANT: The export feature is not a backup tool. The export feature is designed to handle configuration information applicable to all members of a cluster, and network IP addresses and DNS names are filtered out during the import. (The server-specific information that is filtered out is the information you set specifically for each member in a cluster.) If you want a copy of all configuration information, including server-specific information, you need to perform a backup. See Chapter 24, “Back Up and Restore,” on page 925.

The export feature is not an upgrade tool. You cannot export a configuration from one version of Access Manager and import it into a newer version of Access Manager.

If your Access Gateway is not a member of a cluster and you have configured it to use multiple IP addresses, be aware that the export feature filters out multiple IP addresses and uses only eth0. You need to use the backup utility to save this type of information. If you need to reinstall the machine, leave the Access Gateway configuration in the Administration Console and reinstall the Access Gateway. If you use the same IP address for the Access Gateway, it imports into the Administration Console and inherits the configuration.

When exporting the file, you can select to password-protect the file, which encrypts the file. If you are using the exported file to move an Access Gateway from a staging area to a production area and you need to change the names of the proxy services and DNS names from a staging name to a production area and you need to change the names of the proxy services and DNS names from a staging name to a production name, do not select to encrypt the file. You need a simple text file so you can search and replace these names. If you select not to encrypt the file, remember that the file contains sensitive information and protect it accordingly.

Exporting the Configuration

1 In the Administration Console, click **Devices** > **Access Gateway** > **[Name of Access Gateway]**.

2 Click **Configuration** > **Export**.

3 (Conditional) If you want to encrypt the file, fill in the following fields:

Password protect: Select this option to encrypt the file.

Password: Specify a password to use for encrypting the file. When you import the configuration onto another device, you are prompted for this password.

4 Click **OK**, then select to save the configuration to a file.

The filename is the name of the Access Gateway with an `xml` extension.

5 Export the policies used by the Access Gateway. In the Administration Console, click **Policies** > **Policies**, then either select **Name** to include all policies or individually select the policies to export.

You need to export all Access Gateway policies and any Role policies used by the Access Gateway policies.

6 Click **Export** and modify the proposed filename if needed.

7 Click **OK**, then select to save the policy configurations to a file.

8 (Conditional) If you have created multiple policy containers, select the next policy container in the list, and repeat Step 5 through Step 7.

The policies for each container must be saved to a separate export file.

Setting Up a Tunnel

The tunnel option lets you create one or more services for the specific purpose of tunneling non-HTTP traffic through the Access Gateway to a Web server. To do this, the non-HTTP traffic must use a different IP address and port combination than the HTTP traffic.

An Access Gateway usually processes HTTP requests in order to fill them. However, it is not unusual that some of the traffic coming through the gateway is not HTTP-based. Web servers sometimes handle Telnet, FTP, chat, or other kinds of traffic without attempting to process it. If your Web servers are handling this type of traffic, you must set up a tunnel for it.

Reverse proxies and tunnels cannot share the same IP address and port combination. You can either configure a reverse proxy for an IP address and port or a tunnel for that IP address and port.

To set up a tunnel:

1 In the Administration Console, click **Devices** > **Access Gateways** > **Edit** > **Tunneling**.

2 Click **New**, enter a display name for the tunnel, then click **OK**.

3 Specify the following details:

Enable Tunnel: Specifies that the Access Gateway must set up a tunnel for all incoming traffic. This option must be enabled to configure a tunnel.

Tunnel SSL Traffic Only: Allows you to configure the Access Gateway to tunnel only SSL traffic. If this option is selected, the Access Gateway verifies that the address and port being accessed are actually an SSL Web site. If verification fails, the service tears down the connection. The SSL port number for the SSL tunnel is specified via the **Listening Port** and the **Connect Port**.

Published DNS Name: Specify the DNS name you want the public to use to access your tunnel or the virtual IP address assigned to the Access Gateway cluster by the L4 switch. If you specify a DNS name, the DNS name must resolve to the IP address you set up as the listening address for the tunnel.

4 Configure the communication options between the browsers and the tunnel by configuring the following fields:

Cluster Member: (Available only if the Access Gateway is a member of a cluster.) Select the server you want to configure from the list of servers. The **Listening Address(es)** modifications apply to the selected server. Any other modifications apply to all servers in the cluster.

Listening Address(es): Displays a list of available IP addresses. If the Access Gateway has only one IP address, only one is displayed. If it has multiple addresses, you can select one or more addresses to enable. You must enable at least one address by selecting its check box.

TCP Listen Options: Provides additional options for configuring how requests are handled. See "Configuring TCP Listen Options for Clients" on page 131. At least one Web server must be configured before you can modify these options.

Listening Port: Specifies the port on which to listen for requests from browsers. The listening address and port combination must not match any combination you have configured for a reverse proxy.

5 Configure the communication options between the tunnel and the Web servers by configuring the following fields:

Connect Port: Specifies the port that the Access Gateway uses to communicate with the Web server.

TCP Connect Options: Allows you to control how idle and unresponsive Web server connections are handled and to optimize these processes for your network. See "Configuring TCP Connect Options for Web Servers" on page 132.

6 Specify a Web server to receive the traffic. In the Web Server List section, click **New**, specify the IP address or DNS name of the Web server, then click **OK**.

At least one Web server must be specified in the list before you can save a tunnel configuration.

7 To save your changes to browser cache, click **OK**.

8 To apply your changes, click the **Access Gateways** link, then click **Update** > **OK**.

Setting the Date and Time

The **Date & Time** option lets you set the system time for the Access Gateway.

The time between the Identity Server and the Access Gateway must be either synchronized or set to be within 1 minute of each other for trusted authentication to work.

To configure the date and time options:

1 In the Administration Console, click **Devices** > **Access Gateways** > **Edit** > **Date & Time**.

2 (Conditional) If the Access Gateway belongs to a cluster of Access Gateways, select the Access Gateway from the list displayed in the **Cluster Member** field. The modifications you make on this page apply only to the selected Access Gateway.

If the Access Gateway does not belong to a cluster, this option is not available.

3 Specify the following details:

Server Date and Time: Displays the current time and allows you to set the current time. Click **Set Date & Time Manually**, then select the current year, month, day, hour, and minute.

IMPORTANT: If the date is set to a time before the Access Gateway certificates are valid, communication to the Access Gateway is lost. This error cannot be corrected from the Administration Console. You need to correct it at the console of the Access Gateway machine.

Use the `yast` command and select **System** > **Date and Time**.

Set Up NTP: Click this option to specify the DNS name or IP address of a Network Time Protocol server. The installation program enters the name of `pool.ntp.org`, the DNS name of a public NTP server. To disable this feature, you must remove all servers from the NTP Server List. This is not recommended.

Time Zone: Select your time zone, then click **OK**. Regardless of the method you used to set the time, you must select a time zone.

4 Click **OK**.

5 On the Server Configuration page, click **OK**.

6 To apply your changes, click **Update** > **OK**.

Configuring Network Settings

After initial setup, you seldom need to change the network settings unless something in your network changes, such as adding a new gateway or DNS server. These options are for the Access Gateway Appliance. For the Linux or Windows Access Gateway Service, use the utilities supplied by the operating system. However, if you add an new network interface card to the Access Gateway Service machine and use system utilities to configure it and assign it an IP address, you need to update the Access Gateway Service with this information. See “Adding a New IP Address to the Access Gateway” on page 245.

This section describes the following tasks:

- “Viewing and Modifying Adapter Settings” on page 242
- “Viewing and Modifying Gateway Settings” on page 243
- “Viewing and Modifying DNS Settings” on page 244
- “Configuring Hosts” on page 245
- “Adding a New IP Address to the Access Gateway” on page 245

Viewing and Modifying Adapter Settings

The adapter settings allow you to view the current configuration for the network adapters installed in the Access Gateway Appliance and manage the IP addresses that are assigned to them.

- If you want to configure an adapter to use more than one IP address, you can use these settings to add them.
- If you have multiple adapters installed on an Access Gateway Appliance machine, you can only configure eth0 during installation. Use the procedure described in this section to configure the others.

To view or modify your current adapter settings:

1 In the Administration Console, click **Devices** > **Access Gateways** > **Edit** > **Adapter List**.

2 (Conditional) If the Access Gateway is a member of a cluster, select the server you want to configure from the list of servers in the **Cluster Member** field. All changes made to this page apply to the selected server.

3 Select the adapter you want to modify, then select one of the following actions:

- To add a new subnet to an existing adapter, click **New**.
- To delete a subnet, select a subnet, then click **Delete**. More than one subnet must be configured for you to delete one.
- To modify an existing subnet, click the IP address of the subnet.

4 To configure a new subnet or a new IP address for a subnet, configure the following fields:

Subnet: Displays the address of the subnet that you are modifying. This is empty if you are creating a new subnet.

Subnet Mask: (Required) Specifies the subnet mask address for this subnet. The address can be specified in standard dotted format or in CIDR format.

IP Addresses: Allows you to manage the IP addresses assigned to the subnet.

- To add an address, click **New**, specify the address, then click **OK**.
- To delete an address, select the address, then click **Delete**.
- To change the IP address, select the address, then click **Change IP Address**, specify the new IP address, then click **OK**.

5 Click **OK**.

6 Click **OK**.

7 On the Server Configuration page, click **OK**, then click **Update** > **OK**.

Viewing and Modifying Gateway Settings

The gateway settings display the current gateway configuration that the Access Gateway Appliance is using to route packets. On this page, you can also configure additional gateways. During installation, you could specify only a default gateway. You must have at least one gateway defined for the Access Gateway to function.

The Access Gateway routes requests to specific destinations through these gateways. If a request could be routed through multiple gateways, the Access Gateway chooses the gateway associated with the most restrictive mask (the smallest range of destination addresses). The default gateway is used only when no other routes apply.

Gateways fall within the following three basic groups:

- Host gateways for specific destination addresses.
- Network gateways for destination addresses that fall within specific subnets.
- The default gateway for destination addresses that aren't covered by host or network gateways.

The Access Gateway uses additional gateways only when the **Act As Router** option is selected. When this option is selected, you can add Host Gateways and Network Gateways. When configuring a Host Gateway or Network Gateway, you specify the IP address of the host or network gateway in the **Next Hop** field. This address must be on the same subnetwork as the IP address for the Access Gateway.

IMPORTANT: If you enter an IP address that is on a different subnetwork, the Access Gateway reports this error on the Health page, after the configuration has been applied.

To modify your current gateway configuration:

1 In the Administration Console, click **Devices** > **Access Gateways > Edit** > **Gateways**.

2 Configure your default gateway, which specifies the gateway to use when no other routes apply. Configure the following:

Next Hop: The IP address of the gateway.

Metric: A relative number indicating the bias you can add to the normal flow of gateway logic. Specifying a number higher than 1 makes this resource more expensive and alters the gateway logic used. Valid numbers include 1 through 16.

Type: Gateways are active if they publish their presence, or passive if they do not.

3 Configure your host gateways, which are the gateways to be used for packets being sent to specific hosts. When you select **New** from the **Host Gateway** list, you are asked for the following information:

Next Hop: The address of the host gateway that is to be used.

Host: The IP address of the destination host. Valid addresses cannot be the first or last address of a class and must be unique.

Metric: A relative number indicating the bias you can add to the normal flow of gateway logic. Specifying a number higher than 1 makes this resource more expensive and alters the gateway logic used. Valid numbers include 1 through 16.

Type: Gateways are active if they publish their presence, or passive if they do not.

Click **OK** when the fields are configured.

4 Configure your network gateways, which are the gateways to be used for packets being sent to specific subnets. When you select **New** from the **Network Gateway** list, you are asked for the following information:

Next Hop: The address of the gateway that is to be used.

Network Address: The subnet address for the destination IP address range. You must enter the valid subnet address.

Mask: The subnet mask for the subnet or IP address above. A valid entry must be at least as large as a class mask where a Class A mask is 255.0.0.0, a Class B mask is 255.255.0.0, and Class C, D, and E masks are 255.255.255.0.

Metric: A relative number indicating the bias you can add to the normal flow of gateway logic. Specifying a number higher than 1 makes this resource more expensive and alters the gateway logic used. Valid numbers include 1 through 16.

Type: Gateways are active if they publish their presence, or passive if they do not.

Click **OK** when the fields are configured.

5 Click **OK**.

6 On the Server Configuration page, click **OK**, then click **Update** > **OK**.

Viewing and Modifying DNS Settings

The DNS page displays the current configuration for domain name services for the Access Gateway Appliance and allows you to modify it.

1 In the Administration Console, click **Devices** > **Access Gateways** > **Edit** > **DNS**.

2 (Conditional) If the Access Gateway is a member of a cluster, select the server you want to configure from the list of servers in the **Cluster Member** field. All changes made to this page apply to the selected server.

3 Specify the following details:

Server Hostname: Displays the unique host or computer name that you have assigned to the Access Gateway machine. If you modify this name, you need to modify the entry for the Access Gateway in your DNS server to resolve this new name.

Domain: Specifies the domain name for your network. Your DNS server must be configured to resolve the combination of the server hostname and the domain name to the Access Gateway machine. This field assumes you are using dotted names for your machines, such as `sales.mytest.com`, where `sales` is the **Server Hostname** and `mytest.com` is the **Domain**.

DNS Server IP Addresses: Displays the IP addresses of the servers on your network that resolve DNS names to IP addresses. You can have up to three servers in the list. If you specified any addresses during installation, they appear in this list. To manage the servers in this list, select one of the following options:

- **New:** To add a server to the list, click this option and specify the IP address of a DNS server.
- **Delete:** To delete a server from the list, select the address of a server, then click this option.
- **Order:** To modify the order in which the DNS servers are listed, select the server, then click either the up-arrow or the down-arrow buttons. The first server in the list is the first server contacted when a DNS name needs to be resolved.

4 Click **OK**.

5 On the Server Configuration page, click **OK**, then click **Update** > **OK**.

Configuring Hosts

You can configure the Access Gateway Appliance to have multiple hostnames or to resolve DNS names to IP addresses. If you manually edit the `/etc/hosts` file, your modifications are lost when the Access Gateway Appliance is updated. However, if you use the Hosts page to specify the entries, the entries are written to the `/etc/hosts` file whenever the configuration of the Access Gateway Appliance is updated.

1 (Access Gateway Appliance) In the Administration Console, click **Devices** > **Access Gateways** > **Edit** > **Hosts**.

2 (Conditional) If the Access Gateway is a member of a cluster, select the server you want to configure from the list of servers in the **Cluster Member** field. All changes made to this page apply to the selected server.

3 To add a new hostname to an existing IP address, click the name of a **Host IP Address**.

4 In the **Host Name(s)** text box, specify a name for the host. Place each hostname on a separate line, then click **OK**.

5 To add a new IP address and hostname, click **New** in the **Host IP Address List** section, then specify the IP address. In the **Host Name(s)** text box, specify a hostname, then click **OK**.

6 To delete a host, select the check box next to the host you want to delete, then click **Delete**.

7 Click **OK**.

8 On the Server Configuration page, click **OK**, then update the Access Gateway.

Adding a New IP Address to the Access Gateway

Before configuring the Access Gateway to use a new IP address, you must first use an operating system utility to add the IP address.

After you have used a system utility to add an IP address, you need to update the Access Gateway Service to display the new IP address as a configuration option.

1 In the Administration Console, click **Devices** > **Access Gateways** > **[Name of Gateway Service]**.

2 On the Server Details page, click **New IP** > **OK**.

The Access Gateway scans the operating system for its configured IP addresses and adds any new addresses. The new address is then available for assignment on the Access Gateway configuration pages.

3 (Optional) To verify that the scan has completed, click the **Command Status** tab.

Enabling the Access Gateway to Display Post-Authentication Message

When the Identity Server authentication process is completed, the user-agents are redirected to their originally requested URL. The originally requested URL is then retrieved by the proxy. This process requires SSO and authentication process of its own. As a result, retrieving the requested URL may take a long time. It is not clear how much time the authentication process takes and how much time the origin server request and authentication processes take.

To remove this ambiguity, you can enable the Access Gateway to display a message before redirecting the user-agent to the originally requested URL.

To enable this enhancement, complete the following steps:

1. Go to **Devices** > **Access Gateways** > **Edit** > **Reverse Proxy /Authentication** > **ESP Global Options**.
2. Set **IS_DISPLAY_AUTH_DONE_PAGE** to true.

When this option is enabled, the following message is displayed before the final redirect to the requested URL:

```
Authentication successful, please wait while your requested page loads.
```

The web page that display this message is a JSP page. Location of this page is `/opt/novell/nam/mag/webapps/nesp/jsp/waitredir.jsp`. You can perform further customization on this page.

Customizing The Access Gateway

- "Maintaining a Customized Access Gateway" on page 246
- "Customizing Error Messages and Error Pages on Access Gateway" on page 246
- "Customizing Logout Requests" on page 249

Maintaining a Customized Access Gateway

If you have customized the `.jsp` files for the Access Gateway, you must perform the following steps to maintain the customized files before upgrading Access Manager. If you do not, Access Manager overwrites the customized `.jsp` files. For more information, see Maintaining Customized JSP Files for Access Gateway in the NetIQ Access Manager Appliance 4.2 Installation and Upgrade Guide.

Customizing Error Messages and Error Pages on Access Gateway

Access Gateway uses the custom error page template to rebrand and localize the language of error pages that are published to the browser.

By default, Access Gateway contains the following files to help customize and localize the error messages:

- The error page configuration file, `ErrorPagesConfig.xml`
- The error messages file, `ErrorMessages.xml.en`

NOTE: If you are modifying any of the above files, ensure that you retain the original filenames.

Access Gateway maintains `/opt/novell/nam/mag/webapps/agm/WEB-INF/config/current/` directory to save files that are used for error page configuration.

You can customize and localize the error template and the error messages:

- “Customizing and Localizing Error Messages” on page 247
- “Customizing the Error Pages” on page 247

Customizing and Localizing Error Messages

When Access Gateway serves an error message to the browser by using the `Accept-Language` header value received from the browser, it selects a suitable error template and an error message file. To localize the error messages, you must to do the following:

Localize or customize the error messages in the `ErrorPagesConfig.xml` file and save it with the language extension.

The error messages contained in the `ErrorMessages.xml.en` file can be localized in various languages and stored as `ErrorMessages.xml.<`*lang*`>`, where <*lang*> is the `fileXn` attribute value. You can also customize the English error messages present in the `ErrorMessages.xml.en` file.

NOTE: You cannot customize an error message that is not present in the `ErrorMessages.xml.en` file.

To localize the error messages, perform the following steps:

1. Log in as `root`.
2. Open the `ErrorMessages.xml.<lang>` file.
3. Copy the error messages that you have localized or customized to within the `<TranslatedMessage></TranslatedMessage>` tags. For example:

   ```
   </Message>
     <Message id="<ID No>" name="<ERROR_MESSAGE_NAME>" enable="yes">
       <EnglishMessage>English Message goes here</EnglishMessage>
   <TranslatedMessage>
   Localized message goes here
   </TranslatedMessage>
   </Message>
   ```

   Do not delete the contents within the `<TranslatedMessage></TranslatedMessage>` tags from an English file because, the `ErrorPagesConfig.xml` file selects the error message within these tags for display.
4. Save the file.
5. If the Access Gateway belongs to a cluster, copy the modified file to each member of the cluster, then restart that member.
6. Edit the configuration and make dummy changes and push the configuration.

Customizing the Error Pages

Access Gateway uses the Apache method for localizing error messages. You can modify these messages or customize the page they are displayed on.

1. To change a message:

   1a Change to the Apache message configuration directory:

   ```
   /etc/opt/novell/apache2/conf/extra
   ```

1b Open the `http-multilang-errordoc.conf` file.

The first few lines of this file contains comments on how Apache recommends modifying the error messages. You can select to use their method or continue with the following steps.

1c Locate the `ErrorDocument` section and determine the error code message you want to modify. Make note of the `*.var` filename.

1d Change to the Apache error directory:

```
/opt/novell/apache2/share/apache2/error
```

1e Open the `*.var` file that you want to modify.

The message is listed alphabetically by language code.

1f Save the changes.

2 To change the header of the error page:

2a Change to the Apache error include directory:

```
/opt/novell/apache2/share/apache2/error/include
```

2b Open the `top.html` page.

2c To change the title of the page, locate the following line:

```
<title>Access Manager<\title>
```

2d Replace the `Access Manager`string with the content you require.

2e To replace the image in the header, locate the following line:

```
<img src="/NAGErrors/images/header_550.png" alt="" height="50px"
width="550px" border="0">
```

2f Replace `header_550.png` with the filename of the image you want to display.

2g Adjust the height and width values to match your image.

2h Save the file.

2i Copy your image to the `images` directory:

```
/opt/novell/apache2/share/apache2/error/images
```

3 To change the footer of the error page:

3a Change to the Apache error include directory:

```
/opt/novell/apache2/share/apache2/error/include
```

3b Open the `bottom.html` page.

3c To change the image, find the following line:

```
<td style="background-color: #E6D88C; padding-left: 10px"><img
style="padding-right: 200px" src="/NAGErrors/images/
LAP_interoperable_logo_100.gif" align="absmiddle" border="0">
```

3d Change `LAP_interoperable_logo_100.gif` to the filename of the image you want to display.

3e Save the file.

3f Copy your image to the `images` directory:

```
/opt/novell/apache2/share/apache2/error/images
```

4 Copy all modified files and image files to all Access Gateways in the cluster.

The `err_legacy.jsp` file will also log the ESP error messages. For more information on customizing the `err_legacy.jsp` page, see “Customizing Identity Server Messages” on page 211. The procedure for customizing is the same except the paths for the Access Gateway. The following are the path changes:

- In Customizing Identity Server Messages, the paths for Access Gateway are as follows:

  Step 3, path on Linux is `/opt/novell/nam/mag/webapps/nesp/WEB-INF/lib` and on Windows is `/Program Files/Novell/Tomcat/webapps/nesp/WEB-INF/lib/`.

  Step 10, path on Linux is `/opt/novell/nam/mag/webapps/nesp/WEB-INF/classes` and on Windows is `/Program Files/Novell/Tomcat/webapps/nesp/WEB-INF/classes`.

  Step 12, restart the Access Gateway by running `/etc/init.d/novell-mag restart`.
- In Customizing the Branding of the Error Page, the path for `err_legacy.jsp` in the ESP on Linux is `/opt/novell/nam/mag/webapps/nesp/jsp` and on Windows is `/Program Files/Novell/Tomcat/webapps/nesp/jsp/`.

Customizing Logout Requests

- “Customizing Applications to Use the Access Gateway Logout Page” on page 249
- “Customizing the Access Gateway Logout Page” on page 250
- “Configuring the Logout Disconnect Interval” on page 252

Customizing Applications to Use the Access Gateway Logout Page

If any of your protected resources have a logout page or button, you need to redirect the user’s logout request to the Access Gateway logout page. The Access Gateway can then clear the user’s session and log the user out of any other resources that have been enabled for single sign-on. If you do not redirect the user’s logout request, the user is logged out of one resource, but the user’s session remains active until inactivity closes the session. If the user accesses the resource again before the session is closed, single sign-on reauthenticates the user to the resource, and it appears that the logout did nothing.

1. In the Administration Console, click **Devices** > **Access Gateways** > **Edit** > **Reverse Proxy / Authentication**.
2. In the **Embedded Service Provider** section, view the path to the AGLogout page in the **Logout URL** option.

   The Logout URL displays the URL that you need to use for logging users out of protected resources. This option is not displayed until you have created at least one reverse proxy with a proxy service. If you create two or more reverse proxies, you can select which one is used for authentication, and the logout URL changes to match the assigned reverse proxy.
3. Redirect application logout requests to the AGLogout page.
4. Click **OK**.

The Access Gateway does not support the following logout pages that were used in previous version of Access Manager and iChain:

- /cmd/BM-Logout
- /cmd/ICSLogout

Customizing the Access Gateway Logout Page

You can create your own logout page and configure the Access Gateway to use it. To do this, you need to modify the `logoutSuccess_legacy.jsp` file on the Access Gateway. It is located in the following directory:

```
/opt/novell/nesp/lib/webapp/jsp
```

You can modify the file to display what you want or you can modify it to redirect the user to your custom page. The following sections provide some tips for accomplishing this task:

- “Modifying the Header” on page 250
- “Calling Different Logout Pages” on page 250
- “Redirecting to Your Custom Page” on page 251
- “Logging Out of Sessions to the Access Gateway and SAML Connectors when Branding Exists in the Customized Logout Page” on page 252

Modifying the Header

The `logoutSuccess_legacy.jsp` file is called in a frame from the `nidp_legacy.jsp` file. The branding in the header of the logout page is controlled by the branding of the `nidp.jsp` file. For information about how to modify `nidp_legacy.jsp` for logos, titles, and colors, see “Logging Out of Sessions to the Access Gateway and SAML Connectors when Branding Exists in the Customized Logout Page” on page 252.

IMPORTANT: Take a backup of `nidp_legacy.jsp` file before modifications. Every time you upgrade your Access Gateway, upgrade process overrides any custom changes made to JSP files that use the same filename as those included with the product. If you want the modified file, you need to restore the `nidp_legacy.jsp` file. During an upgrade, you can select to restore custom login pages, but NetIQ still recommends that you have your own backup of any customized files.

Calling Different Logout Pages

If you need to use a different logout page for specific protected resources, you can modify the logout button of the applications to use the AGLogout URL or the plogout URL. The AGLogout page calls the plogout page, which in turn calls `logoutSuccess_legacy.jsp`. Any parameter added to the AGLogout or plogout URL is saved and passed to the `logoutSuccess_legacy.jsp` file.

The parameter passed to `logoutSuccess_legacy.jsp` can be used with if/else logic in the body of the page to load different custom logout pages based on the parameter value.

To use the plogout URL, modify the application’s logout button to call the following URL:

```
<ESP Domain>/nesp/app/plogout
```

Replace <*ESP Domain*> with the same value as the AGLogout value. For example, suppose the following is your AGLogout value:

```
https://jwilson1.provo.novell.com:443/AGLogout
```

Replace it with the following value:

```
https://jwilson1.provo.novell.com:443/nesp/app/plogout
```

If you add a parameter to the URL, it would look similar to the following:

```
https://jwilson1.provo.novell.com:443/nesp/app/plogout?app=email
```

Redirecting to Your Custom Page

Server side redirection: You can replace the information in the `<body>` element of the `logoutSuccess_legacy.jsp` file with something similar to the following:

```
<body>
      <script language="JavaScript">
        top.location.href='http://<hostname/path>';
      </script>
</body>
```

Replace the *<hostname/path>* string with the location of your customized logout page.

Client side redirection by using JavaScript: You can customize `logoutSuccess_legacy.jsp` to redirect the logout request to the page that is specified in the query parameter.

For example, add the following logic in `logoutSuccess_legacy.jsp`:

```
<%
    String sImpDomain = request.getParameter("impDomain");
    out.println ("sImpDomain from Request: " + sImpDomain + "<BR>" );

    if (null == sImpDomain)
    {
        sImpDomain = uh.getLogoutQueryStringParam("impDomain");
        out.println ("sImpDomain from Handler: " + sImpDomain + "<BR>" );
    }
    String strDestinationUrl = null;
    if (null != sImpDomain)
    {
            if(sImpDomain.equals("digitalairlines")){
                    strDestinationUrl  = "https://www.DigitalAirlines.com";
            }else if(sImpDomain.equals("example")){
                    strDestinationUrl  = "https://www.example.com";
            }
        if (null != strDestinationUrl)
        {
%>
                <script>
                        alert("Client browser is getting redirected to
<%=strDestinationUrl%>");
                        window.location = "<%=strDestinationUrl%>";
                </script>
<%
        }
    }
%>
```

To test this example, make an AGLogout request from a logged in session in the browser as follows:

`https://<base-url>/AGLogout?impDomain=digitalairlines`

The client browser displays the following message:

`Client browser is getting redirected to https://www.digitalairlines.com`

After clicking **OK**, the browser gets redirected to `https://www.digitalairlines.com`.

Similarly if the AGLogout request is made with `impDomain=example`, then client browser will be redirected to `https://www.example.com`.

IMPORTANT: Take a backup of `logoutSuccess_legacy.jsp` file before modifications. Every time you upgrade your Access Gateway, upgrade process overrides any custom changes made to JSP files that use the same filename as those included with the product. If you want the modified file, you need to restore the `nidp_legacy.jsp` file. During an upgrade, you can select to restore custom login pages, but NetIQ still recommends that you have your own backup of any customized files

Logging Out of Sessions to the Access Gateway and SAML Connectors when Branding Exists in the Customized Logout Page

When you have both Liberty and SAML 2.0 sessions running on the Identity Server and you log out of the Access Gateway, the `logoutSuccess_legacy.jsp` page is not executed with the customization you have made to the logout page. You will be able to log out of the Access Gateway but the customization you made are lost.

If the `logutSuccess_legacy.jsp` file is not loaded in a frame, the banner will not be displayed, and the Access Gateway will comment out the content in the `logoutSuccess_legacy.jsp` file. Add the below line after the <body> tag in the `logoutSuccess_legacy.jsp` file.

```
<!-- BANNER LOADS IF THIS PAGE IS NOT LOADED IN REGULAR FRAME -->

<%@include file="logoutHeader.jsp"%>
```

Configuring the Logout Disconnect Interval

When a user clicks the logout button and the user is logging out of an Access Gateway that is a member of a cluster, the user is not immediately disconnected from the resource. The logout message must be sent to each member of the cluster. The default interval for checking the pending logout message queue is 30 seconds. If this interval is too long, you can configure a shorter interval in the `web.xml` file of the Embedded Service Provider. This must be set on each Access Gateway in the cluster.

1. Log in to the Access Gateway as the root or administrator user.
2. Open `web.xml`.

   `/opt/novell/nesp/lib/webapps/WEB-INF/web.xml`
3. Find the `<context-param>` section in the file.
4. Add the following parameter to the `<context-param>` section.

   ```
   <context-param>
       <param-name>logoutRetirementFrequency</param-name>
       <param-value>15000</param-value>
   </context-param>
   ```
5. Set the `<param-value>` element to a value between 5000 and 30000 milliseconds (5 seconds and 30 seconds).
6. Restart the Embedded Service Provider.

   For information about how to restart the Embedded Service Provider from the Administration Console, see “Managing Access Gateways” on page 234.

Access Gateway Content Settings

One of the major benefits of using an Access Gateway to protect Web resources is that it can cache the requested information and send it directly to the client browser rather than contacting the origin Web resource and waiting for the requested information to be sent. This can significantly accelerate access to the information.

IMPORTANT: For caching to work correctly, the Web servers must be configured to maintain a valid time. If possible, they must be configured to use an NTP server.

The object cache on an Access Gateway is quite different from a browser's cache, which all users access when they click the **Back** button and which can serve stale content that does not' accurately reflect the fresh content on the origin Web server.

The Access Gateway caching system uses a number of methods to ensure cache freshness. Most time-sensitive Web content is flagged by Webmasters in such a way that it cannot become stale unless a caching system ignores the Webmaster's settings. The Access Gateway honors all RFC 2616 directives that affect cache freshness such as Cache-Control, If-Modified-Since, and Expires.

The Access Gateway can be fine-tuned for cache freshness in the following ways:

- Accelerated checking of objects that have longer than desirable Time to Expire headers
- Delayed checking of objects that have shorter than desirable Time to Expire headers
- Checking for freshness of objects that do not include Time to Expire headers

The Access Gateway follow RFC directives. In addition, the Access Gateway Service uses the "Apache Module mod_file_cache" (http://httpd.apache.org/docs/2.2/mod/mod_file_cache.html).

The following sections describe the features available to fine-tune this process for your network:

- "Configuring Caching Options" on page 253
- "Controlling Browser Caching" on page 254
- "Configuring a Pin List" on page 254
- "Configuring a Purge List" on page 257
- "Purging Cached Content" on page 258
- "Apache htcacheclean Tool" on page 258

Configuring Caching Options

The Cache Options allow you to control how the Access Gateway caches objects.

1. Click **Access Gateways** > **Edit** > **Cache Options**.
2. To disable caching of all Web server content, select **Disable Caching**.

   When this option is selected, all other caching options are disabled.
3. Modify the Cache Freshness settings. Use the **Reset** button to return these settings to their default values.

   These options govern when the proxy service revalidates requested cached objects against those on their respective origin Web servers. If the objects have changed, the proxy service re-caches them.

   WARNING: Enter whole number values. Decimal values (2.5) are not supported and generate an XML validation error.

   HTTP Maximum: Specifies the maximum time the proxy service serves HTTP data from cache before revalidating it against content on the origin Web server. No object is served from cache after this value expires without being revalidated.

   This overrides a freshness or Time to Expire directive specified by the Webmasters if they specified a longer time.

You use this value to reduce the maximum time the proxy service waits before checking whether requested objects need to be refreshed. The default is 6 hours.

HTTP Default: Specifies the maximum time the proxy service serves HTTP data for which Webmasters have not specified a freshness or Time to Expire directive. The default is 2 hours.

4 To save your changes to browser cache, click **OK**.

5 To apply the changes, click the **Access Gateways** link, then click **Update** > **OK**.

Controlling Browser Caching

Webmasters control how browsers cache information by adding the following cache-control directives to the HTTP headers:

```
Cache-Control: no-store
Cache-Control: no-cache
Cache-Control: private
Cache-Control: public
Pragma: no-cache
```

You can configure how the proxy service responds to these directives in the HTTP header.

1 In the Administration Console, click **Devices** > **Access Gateways** > **Edit** > **[Name of Reverse Proxy]** > **[Name of Proxy Service]** > **HTTP Options**.

2 To mark all pages coming through this host as cacheable on the browser, select **Allow Pages to be Cached by the Browser**.

When this option is enabled, the no-cache and no-store headers are not injected into the HTTP header.

You need to select this option if you have a back-end application that updates the data in the Last-Modified or ETag HTTP headers. These changes are forwarded from the Web server to the browser only when this option is enabled.

You need to select this option if you want the Expires HTTP header forwarded from the Web server to the browser.

If this option is not selected, all pages are marked as non-cacheable on the browser. This forces the browser to request a resend of the data from the Access Gateway when a user returns to a previously viewed page.

3 Click **OK**.

4 To apply the changes, click the **Access Gateways** link, then click **Update** > **OK**.

Configuring a Pin List

A pin list contains URL patterns for identifying objects on the Web. The Access Gateway uses the list to prepopulate the cache, before any requests have come in for the content. This accelerates user access to the content because it is retrieved from a local cache rather than from an exchange with the Web server, which would read it from disk.

You can use the pin list to specify the following:

- Which objects you want to cache
- Which objects you never want cached

The pin list is global to the Access Gateway and affects all protected resources. The objects remain in cache until their normal cache limits are reached or they are bumped out by more recently requested objects.

To configure a pin list:

1. In the Administration Console, click **Devices** > **Access Gateways** > **Edit** > **Pin List**.
2. Select the **Enable Pin List** option to enable the use of pinned objects. If this option is not selected, the pinned objects in the pin list are not used.
3. In the **Pin List** section, click **New**.
4. Fill in the following fields.

   URL Mask: Specifies the URL pattern to match. For more information, see "URL Mask" on page 255.

   Pin Type: Specifies how the URL is to be used to cache objects. Select from **Normal** and **Bypass**. For more information, see "Pin Type" on page 257.
5. To save the list item, click **OK**.
6. To save your changes to browser cache, click **OK**.
7. To apply the changes, click the **Access Gateways** link, then click **Update** > **OK**.

URL Mask

The URL mask can contain complete or partial URL patterns. A single URL mask might apply to a large set of URLs, or it might be so specific that only a single file on the Web matches it.

The Access Gateway processes the masks in the pin list in order of specificity. A mask containing a hostname is more specific than a mask that specifies only a file type. The action taken for an object is the action specified for the first mask that the object matches.

The Access Gateways recognizes four levels of specificity, using the following format:

Level	Examples

hostname	`http://www.foo.gov/documents/picture.gif`
	`http://www.foo.gov/documents/*`
	`http://www.foo.gov`
	`foo.gov/documents/*`
	`foo.gov/*`
	All of these are classified as hostnames, and they are ordered by specificity. The first item in the list is considered the most specific and is processed first. The last item is the most general and is processed last.

Level	Examples
path	/documents/picture.gif /documents/pictures.gif/* /documents/* Path entries are processed after hostnames. A leading forward slash must always be used when specifying a path, and the entry that follows must always reference the root directory of the Web server. In these examples, documents is the root directory. The /* at the end of the path indicates that the entry is a directory. Its absence indicates that the entry is a file. In these examples, picture.gif is a file and pictures.gif/* and documents/* are directories. If you enter a path without the trailing *, the path matches only the directory. With the trailing *, the path matches everything in the directory and its subdirectories. These path entry examples are ordered by specificity. The objects in the /documents/picture.gifdirectory are processed before the objects in the /documents directory.
filename	/picture.gif /widget.js /widget.jp*g /picture*group.gif /DailyTask /DailyTask* Filenames are processed after paths. A leading forward slash must always be used when specifying a filename. You can add asterisks in the file names.
file extension	/*.gif /*.js /*.htm File extensions are processed last. They consist of a leading forward slash, an asterisk, a period, and a file extension.

NOTE: More than one wildcard is not allowed in a URL mask. For example, /*picture.g*f is not correct.

Also, the wildcard must be only in the last part of the path. For example:

Correct: /picture/*.gif

Incorrect: /documents/*/picture.gif

Specific rules have precedence over less specific rules. Thus, objects matched by a more specific rule are always processed according to its conditions. If a less specific rule also matches the object, the less specific rule is ignored for the object. For example, assume the following two entries are in the pin list:

URL Mask	Pin Type	Pin Links
`http://www.foo.gov/documents/*`	normal	1
`www.foo*`	bypass	N/A

The first entry, because it is most specific, caches the pages in the `documents` directory and follows any links on those pages and caches the linked pages. The second entry does not affect what the first entry caches, but it prevents any other domain extensions such as `.com`, `.net`, or `.org` whose DNS names begin with `www.foo` from being cached.

Pin Type

The pin type specifies how the Access Gateway caches objects that match the URL mask.

- **Normal:** The Access Gateway handles objects matching the mask in the same way it handles any other requested objects. In other words, the objects are cached but not pinned.

  Administrators often use this pin type in combination with a broad URL mask that has a bypass pin type. This allows them to insulate specific objects from the effects of the bypass rule.

  For example, you could specify a URL mask of `/*.jpg` with a pin type of bypass and a second URL mask of `www.foo.gov/graphics/*` with a pin type of normal. This causes all files, including `.jpg` files, in the graphics directory on the `foo.gov` Web site to be cached as requested. Assuming there are no other URL masks in the pin list, all other JPG graphics are not cached because of the `/*.jpg` mask.
- **Bypass:** The Access Gateway does not cache the objects. In other words, you can use this option to prevent objects from being cached.

Configuring a Purge List

The purge list is global to the Access Gateway and affects all protected resources. This option allows you to specify URL patterns or masks for the pages and sites whose objects you want to purge from cache.

When you specify the URL mask, do not specify a port. Ports are not stored in the cache file that is used to match the URLs that must be purged.

When defining the masks, keep in mind that the Access Gateway interprets everything in the URL mask between the asterisk wildcard (*) and the following delimiter as a wildcard. Delimiters include the forward slash (/), the period (.), and the colon (:) characters. For example:

URL Mask	Effects
/*.pdf	Causes all PDF files to be purged from cache.
www.foo.gov/contracts/*	Causes all objects in the `contracts` directory and beyond to be purged from cache.

This option also allows you to purge cached objects whose URL contains a specified query string or cookie. This mask is defined by placing a question mark (?) at the start of the mask followed by text strings and wildcards as necessary. String comparisons are not case sensitive. For example, `?*=SPORTS` purges all objects with the text `=SPORTS` or any other combination of uppercase and lowercase letters for `=SPORTS` following the question mark in the URL.

IMPORTANT: If you also configure a pin list, carefully select the objects that you add to the pin and purge lists. Make sure you don't configure a pin list that adds objects to the cache and a purge list that removes the same objects.

1. In the Administration Console, click **Devices** > **Access Gateways** > **Edit** > **Purge List**.
2. Click **New,** enter a URL pattern, then click **OK**.
3. (Optional) Repeat Step 2 to add additional URL patterns.
4. To save your changes to browser cache, click **OK**.
5. To apply the changes, click the **Access Gateways** link, then click **Update** > **OK**.

Purging Cached Content

You can select to purge the content of the purge list or all content cached on the server.

1. In the Administration Console, click **Devices** > **Access Gateways**.
2. Select the name of the server, then click **Actions**.
3. Select one of the following actions:

   Purge List Now: Click this action to cause all objects in the current purge list to be purged from the cache.

   Purge All Cache: Click this action to purge the server cache. All cached content, including items cached by the pin list, is purged.
4. Click either **OK** or **Cancel**.

When you make certain configuration changes such as updating or changing certificates, changing the IP addresses of Web servers, or modifying the rewriter configuration, you are prompted to purge the cache. The cached objects must be updated for users to see the effects of such configuration changes. If your Access Gateways are in a cluster, you need to manage the purge process so your site remains accessible to your users. You must apply the configuration changes to one member of a cluster. When its status returns to healthy and current, issue the command to purge its cache. Then apply the changes to the next cluster member.

IMPORTANT: Do not issue a purge cache command when an Access Gateway has a pending configuration change. Wait until the configuration change completes.

Apache htcacheclean Tool

If you have caching issues with inodes, disk space, and cache corruption in the Access Gateway, use Apache htcacheclean tool which is used to keep the size of mod_disk_cache's storage within a certain limit. This tool can run either manually or in daemon mode. When running in daemon mode, it sleeps in the background and checks the cache directories at regular intervals for cached content to be removed.

The htcacheclean utility tool is located at:

On Linux: /opt/novell/apache2/sbin

The default cache location is:

On Linux: `/var/cache/novell-apache2`

Example: To clear 1024 MBytes of cache, run the following command:

On Linux: `./htcacheclean -v -t -p/var/cache/novell-apache2 -l1024M`

For more information about Apache htcacheclean tool, see htcacheclean - Clean up the disk cache (https://httpd.apache.org/docs/2.2/programs/htcacheclean.html).

Access Gateway Advanced Options

The following sections describe the advanced options along with examples:

- “Configuring the Global Advanced Options” on page 259
- “Configuring the Advanced Options for a Domain-Based and Path-Based Multi-Homing Proxy Service” on page 269

Configuring the Global Advanced Options

The following settings apply to all reverse proxies, unless the option is overwritten by an advance proxy service setting (see “Configuring the Advanced Options for a Domain-Based and Path-Based Multi-Homing Proxy Service” on page 269). The advanced options are disabled by default and will be enabled when they are added.

Advanced Access Gateway Options

1. In the Administration Console, click **Devices** > **Access Gateways** > **Edit** > **Advanced Options.**
2. To activate these options, configure the value, save your changes, then update the Access Gateway. To deactivate these options, add the pound (#) symbol.

The following table lists the advanced options along with their descriptions, default values, and examples. Most of the global advanced options are prefixed with `NAGGlobalOptions` and the domain-based and path-based multi-homing proxy service options have been identified and mentioned below the option name.

Table 4-1 *Advanced Access Gateway Options*

Advanced Option	Description
`NAGGlobalOptions FlushUserCache=on` This is a global advanced option.	Specifies whether cached credential data of the user is updated when the session expires or the user changes an expiring password. This option is equivalent to `PasswordMgmt` in the 3.1 SP4 Access Gateway Appliance. ◆ When this option is on, which is the default setting, the credentials and the Identity Injection data are refreshed. ◆ When this option is turned off, the cached user data can become stale. For example, if your password management service is a protected resource of the Access Gateway and this option is turned off, every time a user changes an expiring password, the user’s data is not flushed and the Access Gateway continues to use stale data for that user.
`NAGGlobalOptions DebugHeaders=on`	When this option is enabled, an X-Mag header is added with debug information. The information can be seen in sniffer traces and with plug-ins such as ieHTTPHeaders, Live HTTP Headers, and FireBug. This option must only be enabled when you are working with NetIQ Support and they instruct you to enable the option.

Advanced Option	Description
`NAGGlobalOptions DebugFormFill=on` This is a global advanced option.	When this option is enabled, additional debug information related to the processing of a Form Fill policy is added to the Apache error log files (`error_log` file under `/var/log/novell-apache2` and to the X-Mag header in the response to browser. The Form Fill entries generated by this option begin with a `FF:` marker. For example, `Oct 23 12:38:29 mag326 httpd[29345]: [warn] AMEVENTID#36: FF:fillSilent: kfh5ummigbq6uGeneral_SS_non_SS_autosumit_Page_13310, referer: https://www.idp.com:8443/nidp/idff/sso?sid=0 Oct 23 12:38:29 mag326 httpd[29345]: [warn] AMEVENTID#36: FF:fillInplaceSilent: kfh5ummigbq6uGeneral_SS_non_SS_autosumit_Page_13310, referer: https://www.idp.com:8443/nidp/idff/sso?sid=0`
`NAGGlobalOptions ESP_Busy_Threshold=<value>` This is a global advanced option.	Proxy starts sending errors to the browser if ESP's average response time in the last one minute is more than the specified value (time in milliseconds).
`NAGGlobalOptions noTOPR` This is a global advanced option.	Disables the activity based time-out in proxy. The proxy redirects browser requests after soft timeout of configured timeout value. This option is equivalent to `disabletoppr` in the 3.1 SP4 Access Gateway Appliance.
`NAGGlobalOptions InPlaceSilent=on` This is a global advanced option.	This enables single sign on to certain Web sites that require the login page to remain as is without any modifications to its structure. If you are using this advanced option for a Form Fill on a page with multiple forms, by default, the first form is posted. If you want to post forms other than the first form, use `NAGGlobalOptions InPlaceSilentPolicyDoesSubmit=on`. For more information, refer to TID 7011817. This option is equivalent to `.enableInPlaceSilentFill` in the 3.1 SP4 Access Gateway Appliance.
`NAGGlobalOptions ForceUTF8` This is a global advanced option.	When this option is enabled, the Access Gateway uses the UTF-8 character set to serve the Form Fill page to the browser. This option is equivalent to `forceUTF8Charset` in the 3.1 SP4 Access Gateway Appliance.
`NAGGlobalOptions AllowMSWebDavMiniRedir=on` This is a global advanced option.	This option helps the user to disable the following functionality, which is enabled by default. If a Microsoft Network Places client sends an `OPTIONS` request with `MS-WebDAV-MiniRedir useragent` to the Access Gateway, then it receives 409 conflict response. The client uses this response to change the user agent to `MS Data Access Internet Publishing Provider DAV`. This option is equivalent to `AllowMSWebDavRedir` in the 3.1 SP4 Access Gateway Appliance.
`NAGGlobalOptions noURLNormalize=on` This is a global advanced option.	When this option is enabled, it disables the URL normalization protection for back-end Web servers. This option resolves issues in serving Web content from Web servers that have double-byte characters such as Japanese language characters. By default, this option is set to off and URL is normalized before sending it to back end Web server.

Advanced Option	Description
`NAGAdditionalRewriterScheme webcal://` This is a global advanced option.	When this option is enabled, the rewriter rewrites URLs that have a scheme of `webcal://`. The default rewriter configuration only rewrites URLs with a scheme of `http://` or `https://`.
`NAGGlobalOptions AppendProviderID=on` This is a global advanced option.	When this option is enabled, it displays the ESP Provider ID in the Access Gateway authorization audit logs. This option helps to know the issues related to ESP provider ID in the audit log file.
`NAGGlobalOptions InPlaceSilentPolicyDoesSubmit=on` This is a global advanced option.	This option must be used to fill forms with complex JavaScript or VBScripts. This option is equivalent to `.enableInPlaceSilentFillNew` in the 3.1 SP4 Access Gateway Appliance.
`NAGGlobalOptions NAGErrorOnIPMismatch=off` This is a global advanced option.	If the value for this option is set to off, the Access Gateway does not perform the IP address check on incoming session cookies. Use this in a setup where two L4 switches are configured in parallel and the browser requests are bounced between these L4 switches. This option is equivalent to `.lagdisableAuthIPCheck` in the 3.1 SP4 Access Gateway Appliance. For example, if multiple back-end Web servers are accelerated by the Access Gateway, some users complain that they are not able to complete logging in. When they access the protected resources, they are redirected to the Identity Server for authentication, but they are not redirected to the original URL. If multiple paths (at the network level) exist between a browser and the Access Gateway and proxies or NAT devices exist on these paths, it is possible that the source IP address of the incoming requests into the Access Gateway might change. For example, assume that user A connects to an ISP. This ISP has multiple transparent proxies in parallel for performance reasons. User A accesses the Access Gateway for the first time. The request from User A goes through a local transparent proxy TP1, so the incoming IP address of the initial request has that transparent proxy's (TP1) IP address. The Access Gateway session cookie is set and the user is redirected back to the page the user was going to originally. User A then sends the next request for this original page, but it goes through a different proxy, TP2. The incoming IP address of the request into the Access Gateway is now different than the one that the user used for authentication (TP1 IP address) and the validation fails. The Access Gateway loops as it continues to request the user to send a valid session cookie. **NOTE:** On receiving IPC cookie from browser, the Access Gateway checks for the client IP address in the cookie. If the IP address in the cookie and the client IP address from which the request came do not match, Access Gateway displays an error page.

Advanced Option	Description
`NAGGlobalOptions NAGDisableExternalRewrite=on` This is a global advanced option.	Access Gateway does not insert the path for the links with external published DNS when you enable this option. This option is equivalent to `.disableExternalDNSRewrite` in the 3.1 SP4 Access Gateway Appliance.
`DisableGWSHealth on` This is a global advanced option.	When this option is enabled, Access Gateway does not check health of the Web server with the back-end server. This option is equivalent to `.disableWSHealth` in the 3.1 SP4 Access Gateway Appliance.
`NAGIchainCookieVersion on` This is a global advanced option.	When this option is enabled, Access Gateway sends the proxy session cookie to the back-end server as IPCZQX01<clusterid>.
`IgnoreDNSServerHealth on` This is a global advanced option.	When this option is used, the Access Gateway does not send the DNS server health status when the Access Gateway health is reported to the Administration Console. When you set the option to `IgnoreDNSServerHealth off <`*`lookupname`*`>`, the Access Gateway sends a DNS query with the specified <*lookupname*>. The Access Gateway sends a successful message to the Administration Console if it connects to the DNS server, else it will send an unable to connect message. By default if you have not specified any option, the Access Gateway sets the option as `IgnoreDNSServerHealth off www.novell.com`. This option is equivalent to `ignoreDnsServerHealth` in the 3.1 SP4 Access Gateway Appliance.
`NAGHostOptions primaryWebdav=<path of pbmh service>` This is a global advanced option.	This option enables users who use the Microsoft Network Places client to connect to the WebDAV folders of a SharePoint server when the SharePoint server has been configured as a path-based multi-homing service on the Access Gateway. This must be added to master proxy service **Advanced Options** whose path based child services accelerates webdav resources with remove path on fill option enabled. This option is equivalent to `.modifyRequestURI` in the 3.1 SP4 Access Gateway Appliance.
`NAGGlobalOptions NAGRenameCookie=on` This is a global advanced option.	Set this option to off to prevent the session ID from getting changed automatically.By default, this option is set to on
`NAGHostOptions mangleCookies=on` This is a proxy option.	This option invalidates the cookies set by the Web server when the user logs out of Access Manager. By default, the Access Gateway does not mangle the cookies that are sent by the web server. Proxy mangles the cookies that are sent by the web server using the user information and sets these mangled cookies at the browser. When a browser sends the mangled cookies to proxy, it de-mangles them using the user information and sends the de-mangled cookies to the web server. For more information about this option, see “Enabling Cookie Mangling” on page 268.

Advanced Option	Description
`AGWSMangleCookiePrefix` This is a proxy option.	Use the `NAGWSMangleCookiePrefix <AnyString>` option to specify the string added to the application cookie after manipulation. For more information about this option, see "Enabling Cookie Mangling" on page 268.
`NAGHostOptions webdavPath=/_vti_bin` This is a global advanced option.	This can be added to master proxy service **Advanced Options** which path based child services with **remove path on fill** option enabled accelerating webdav resources.
`NAGChildOptions WebDav=<path of pbmh service>` This is a global advanced option.	This option can be added to any path based service that accelerates webdav resources with **remove path on fill** enabled. This option is equivalent to `.modifyRequestURI` in the 3.1 SP4 Access Gateway Appliance.
`EnableWSHandshake on` This is a global advanced option.	Setup a firewall between the Access Gateway and the back-end Web server. When the Access Gateway performs heartbeat check with a simple TCP connect to the Web server, the Web server may throw a TLS handshake error. This may cause the firewall, after a certain threshold, to block the connection.This option enables the Access Gateway to perform a SSL handshake while performing a heartbeat check on the back-end SSL-enabled Web server so that the Web server does not respond with a TLS handshake error. By default, the Access Gateway performs a simple TCP connect while performing a heartbeat check on the back-end Web server.
`NAGGlobalOptions IIRemoveEmptyHeaderValue` This is a global advanced option.	This option enables the Identity Injection policy not to send an empty header with null value when a value is not available. By default, the Access Gateway sends an empty header with a null value if a value is not available. For example, applications may have a public and a protected resource configured. Both resources may use an identity injection policy such as to inject an USERID. The public resource uses the user name if authenticated. If the user accesses the public resource (before authentication), the Access Gateway sends an empty header variable USERID. Web servers may not handle an empty header and may respond with an error. In such a scenario use the advanced option to stop the Access Gateway from sending an empty header with null value.

`DumpHeaders on`
 	
`DumpHeadersFacility user`
 	
This is a global advanced option.
	These options ensure that the proxy, logs the user headers to `/var/opt/novell/nam/logs/mag/apache2/error_log` file.

`NAGFilteroutUrlForAudit` This option is available for both domain-based and path-based multi-homing proxy services.	You can add this option to proxy service that filters out specific URLs from auditing (URL Accessed). For example, `NAGFilteroutUrlForAudit ".*.jpg"`, and `NAGFilteroutUrlForAudit ".*.gif"`.

Advanced Option	Description
`SSLProxyVerifyDepth=3` This is a global advanced option.	Specifies how many certificates are in a Web server certificate chain. When you activate the verification of the Web server certificate with the **Any in Reverse Proxy Trust Store** and the public certificate is part of a chain, you need to specify the number of certificates that are in the certificate chain. For more information about configuring Web servers for SSL, see "Configuring SSL between the Proxy Service and the Web Servers" on page 828. ◆ The default search level that is when the attribute **SSLProxyVerifyDepth** is commented to1, if the number of certificates in the Web server certificate chain is greater than 1, then the **SSLProxyVerifyDepth** option must be enabled and must be assigned to the respective value (equal to the number of certificates in the chain).
`ProxyErrorOverride` This is a global advanced option.	Allows you to specify which errors you want returned to the browser unchanged by the Gateway Service. The default behavior of the Gateway Service is to replace Web server errors with Gateway Service errors. However, some applications put more information, such as keys and JavaScript in the message. If this information is critical, specify an override and allow the error message to be returned to the browser without any modifications. For example, NetStorage requires an override for the 401 error because it includes a key in the 401 error. The portal page for the Novell Open Enterprise Server requires an override for error 403 because it includes JavaScript. You can use the following syntax to set this option: ◆ `ProxyErrorOverride on -401 -403`: Allows all errors to be changed to Gateway Service errors except errors 401 and 403, which are sent unchanged. This syntax allows you to list the few errors you want to forward without change while allowing all the others to be changed to Gateway Service errors. ◆ `ProxyErrorOverride off +401 +403`: Disables the changing of Web server errors to Gateway Service errors except for errors 401 and 403, which are changed to Gateway Service errors. Use this option when you have only a few errors that you want changed to Gateway Service errors. **NOTE:** Enable the error codes 401 and 403 for override if you are using Identity Manager 4.0 with Role Mapping Administrator.
`CacheIgnoreHeaders` This option is available only for domain-based proxy service.	Prevents the Access Gateway from writing any Authorization headers to disk. This option is enabled by default, because writing Authorization headers to disk is a potential security risk. You can allow Authorization headers to be written to disk by placing a pound (#) symbol in front of the option or by setting it to `None`. For more information about this Apache option, see "CacheIgnoreHeaders Directive" (http://httpd.apache.org/docs/2.2/mod/mod_cache.html#cacheignoreheaders). **NOTE:** All the path-based services under the domain-based service will inherit the new value.

Advanced Option	Description
`CacheMaxFileSize` This option is available only for domain-based proxy service.	Configuring this option in the **Advanced Options** of a proxy service allows you to set the size of the file that can be stored in the cache. By default the size is set to 5 MB. Add the line CacheMaxFileSize <*bytes*>, for example, CacheMaxFileSize 99900000. **NOTE:** All the path-based services under the domain-based service will inherit the new value.
`NAGErrorOnDNSMismatch` This is a global advanced option.	If SSL is not enabled in reverse proxy, an error message stating **Host Name Does Not Match** is displayed.
`NAGChildOptions WebDav=/Path` This option is valid only for path-based multi-homing proxy service.	Allows the proxy service to handle the specified path. Remove the pound (#) symbol and replace */Path* with the path you want the proxy service to handle.

`SSLHonorCipherOrder`
 	
This is a global advanced option.
	This option enables you to customize the `SSLCipherSuite` used by the Access Gateway. This helps you in taking preventive measures when new vulnerabilities are published.
	
	To avoid Browser Exploit Against SSL/TLS (BEAST) attacks, use the advanced option as follows:
	
	`SSLHonorCipherOrder on`
	
	`SSLCipherSuite !aNULL:!eNULL:!EXPORT:!DSS:!DES:RC4-SHA:RC4-MD5:ALL`

`NAGGlobalOptions onFormFillPolicyRedirUseHttp=on` This is a global advanced option.	This option enables Access Gateway to redirect based on HTTP status code 302 along with the location header when Form Fill policy requires redirect. By default, Access Gateway uses JavaScript to trigger redirect in Form Fill policy. You can use this advanced option when there are issues with JavaScript redirects.
`NAGLAGCompatiability on`	This option enables sharing of session information between the 3.1 SP4 Access Gateway Appliance and the 4.0 Access Gateway Appliance during the process of migration. This option is added by default during the process of migration to ensure communication between the two appliances. You can disable or remove this option after the migration is complete.
`ProxyPassIgnorePathCase on`	Use this option to make the path-based multi-homing path URL case-insensitive. For example, if you have set up a path based proxy /profile in Administration Console and the end user wants to access the URL https://www.lagssl.com/Profile/Security/login.aspx and not https://www.lagssl.com/profile/Security/login.aspx. By default url path is case sensitive.
`NAGPostParkingSizeInKiloBytes`	This option allows you to change the post data parking size limit if an error occurs after you post large data (more than 56 KiloBytes in size) after a session timeout.
`NAGSendURLinErrorResponses on`	This option will not include a href when you access a protected resource and a 302 redirect occurs.

Advanced Option	Description
`SSLProtocol`	This option is supported by the Access Gateway when listening as a server to clients (typically browsers). This directive specifies SSL protocols for mod_ssl to use when establishing the server environment. Clients can only connect with one of the specified protocols. The accepted values are SSLv3, TLSv1, TLSv1.1, TLSv1.2 and all of these. The syntax for this is `SSLProtocol [+-]protocol`. For example, SSLProtocol +SSLv3. For more information about configuring the SSL versions, see Apache documentation.
`SSLProxyProtocol`	This option is supported by the Access Gateway when listening as a server to clients (typically browsers). This directive specifies SSL protocols for mod_ssl to use when establishing a proxy connection in the server environment. Proxies can only connect with one of the specified protocols. The accepted values are SSLv3, TLSv1, TLSv1.1, TLSv1.2 and all of these. The syntax for this is `SSLProxyProtocol [+-]protocol`. For example, SSLProxyProtocol +SSLv3. For more information about configuring the SSL versions, see Apache documentation.
`NAGSessionKey Default` This is a global advanced option.	For additional security in case of cross-domain authentication, the Access Gateway session Cookie is encrypted before sending it as a URL query parameter. For example: In earlier releases of Access Manager, the URL is: `https://novell.blr.com:9443/ -CIPCZQX03218a425f=01000300a463892f582b51722510f334a4223149` In Access Manager 4.1, the URL is: `https://novell.blr.com:9443/%20-CECCjdOOBPIqZZNtF+dRlAyDfTFvOPwnO0xzOQTcnrubNzJ6GFe6FF8dWRzzg7RY9iZJYxNLaU80KnJOoqtqf6u2g==` This advanced option `NAGSessionKey` can be used to specify the password as per the administrator's needs. Passwords with more characters increase the strength of the password and therefore leads to better security. For example: `NAGSessionKey NAM-CROSS-DOMAIN-SESSION-KEY-ENCRYPTION-PASSWORD.` By default, the password is set to "default".
For Windows: `SSLProxyCACertificateFile "C:\ProgramFiles\Novell\apache\cacerts\myserver.pem"` For Linux: `SSLProxyCACertificateFile /opt/novell/apache2/cacerts/myserver.pem`. This is a service level advanced option.	This option prevents failure in SSL connection between Access Gateway and webserver, when a self-signed certificate is used. To prevent this, import the webserver certificates to the proxy trust store. After importing, the webserver certificates, use this advanced option.

Advanced Option	Description
`AllowEncodedSlashes NoDecode` This is a global advanced option.	When this option is enabled, URLs are accepted, but encoded slashes are not decoded. For example, the server accepts the encoded URL `www.example.com%2Ffinance`, but does not try to decode the encoded slash (%2F for /).
`AdditionalBalancerMember Options`	The proxy server checks the web server for each new session request at an interval of one minute by default. You can configure this advanced option to specify a different interval. For example, specify `AdditionalBalancerMemberOptions retry=180`, where 180 is in seconds. You can set the following parameters for this option: ◆ min ◆ max ◆ smax ◆ acquire ◆ connectiontimeout ◆ disablereuse ◆ flushpackets ◆ flushwait ◆ ping ◆ loadfactor ◆ redirect ◆ retry ◆ status For more information about these parameters, see Apache Module mod_proxy. The following parameters are not supported: `keepalive`, `lbset`, `route`, `timeout`, and `ttl`
`RWOutboundHeaderQueryStr ing on` This is a proxy level option.	This option enables outbound header query string rewriting.
`NAGAddProxyHeader` on This is a service level advanced option.	When this option is set to off, Access Gateway will not send the XForwarded Headers to the back-end web server. By default, this option is set to on.
`NAGGlobalOptions ExcludeDNSFull` on This is a global advanced option.	When this option is set to on, the DNS name is excluded from being rewritten by that domain. The HTML Rewriting does not happen when the backend DNS name is included in the **Exclude DNS Name list**.

Options to Clean Up Thick Client State At Browser

Currently, when the idle timeout is detected by the Access Gateway, the user is redirected to the Identity server for authentication. If the content type and url pattern used by the client (as defined in the advanced options `NAGUrlPattern` and `NAGContentType`), the user must be redirected to a pre-defined timeout as defined in the `NAGAuthFrontChannel` advanced option.The redirected URL will also have additional information like the ESP login URL, the contract name as well as the landing page URL as defined in the advanced options. The following advanced options must be used together to clean up the thick client.

Advanced Option	Description
`NAGLauncher`	URL that launches the client.
`NAGUrlPattern /messagebroker/*`	URL pattern that identifies if a specific request came from a client.
`NAGContentType application/x-amf`	Content type in the Request header that is used to identify if the request is a client.
`NAGAuthBackChannel /namtimeout/timeoutamf`	Timeout handler on the server.
`NAGAuthFrontChannel`	Timeout handler on the server which includes the published DNS name of the server.

Enabling Cookie Mangling

When you log out of Access Manager, the Access Manager session cookies will be invalidated on all Identity and Access Gateway Servers. However, the application session cookie is left unchanged on both the browser and the origin Web server. If a different user authenticates to Access Manager again on the same browser and accesses the proxied Web server, the browser may resume the previously established HTTP session with the Web server so that the new user inherits the old logged out users session. The Cookie Mangling feature in Access Gateway prevents this scenario from occurring by manipulating the application cookies set by the Web servers, and invalidating these cookies when the user logs out of Access Manager.

The two advanced Access Gateway options required to enable this functionality are the `NAGHostOptions mangleCookies` and `NAGWSMangleCookiePrefix`. By default, the option `NAGHostOptions mangleCookies` is set to Off.

To enable this feature, add the options, `NAGHostOptions mangleCookies=on` and `NAGWSMangleCookiePrefix <AnyString>` in the Global Advanced Option.

Use the `NAGWSMangleCookiePrefix <AnyString>` option to specify the string added to the application cookie after manipulation. You can replace *<AnyString>* with a string of your choice. For example, adding the `NAGWSMangleCookiePrefix AGMANGLE` results in the `Set-Cookie: AGMANGLEa50b_DzkN=5a8G0` application level cookie set in the browser.

URL Attribute Filter

This feature lets you define filtering options for each proxy service. It helps in filtering out specified URLs from the ones audited as part of the URL Accessed audit event. These filtered out URLs will not be displayed in the Audit Server. This is helpful where auditing every URL is not required and may increase the load on the Audit Server. Unnecessary URLs for example, public images, public javascripts, css and favicons can be safely ignored from auditing. The option to set this feature is

`NAGFilteroutUrlForAudit <regular expression>`. This option must be added to the Advanced options section of each service. The regular expression is standard perl based regular expressions. For more information, see Regular Expressions (http://perldoc.perl.org/perlre.html).

Each URL (path?querystring) is matched against this expression. If the match is successful, the URL will not be audited for URL access. For example, `NAGFilteroutUrlForAudit ".*.jpg"` and `NAGFilteroutUrlForAudit ".*.gif"`. If these options are added to a service, all the `*.jpg` and `*.gif` files accessed will not be audited under the 'URL Accessed' audit event.

NOTE: If you enable 'URL Accessed' audit events in the Access Gateway, it can overload the Audit subsystem if the requests sent to the Gateway per second is high. There maybe a delay in Web pages getting loaded. NetIQ recommends to use the `http common/extended logging` option for this purpose.

Configuring the Advanced Options for a Domain-Based and Path-Based Multi-Homing Proxy Service

The following procedure helps you configure the advanced options for domain-based and path-based multi-homing proxy service of an Access Gateway.

1. In the Administration Console, click **Devices** > **Access Gateways** > **Edit** > **[Name of Reverse Proxy]** > **[Name of Proxy Service]** > **Advanced Options**.
2. Configure the advanced option by removing the pound(#) symbol. To disable an option, add the # symbol in front of the option, save your changes, then update the Access Gateway.

Modifying Configuration Files

You can modify configuration files such as server.xml and web.xml to set up various configurations.

Modifying web.xml

You can modify the `web.xml` file to perform the following configurations:

- Managing the Administration Console Session Timeout. For information about how to configure it, see “Managing the Administration Console Session Timeout” on page 39.
- Configuring the Logout Disconnect Interval. For information about how to configure it, see “Configuring the Logout Disconnect Interval” on page 252.
- Preventing Error Messages to Show the Failure Reason on Browsers, For information about how to configure it, see “Preventing Error Messages to Show the Failure Reason on Browsers” in the *NetIQ Access Manager 4.2 Best Practices Guide*.
- Disabling User Profile Objects. For information about how to configure it, see “Disabling User Profile Objects” in the *NetIQ Access Manager 4.2 Best Practices Guide*.
- Configuring a Specific IP Address for Proxied Requests. For information about how to configure it, see “Configuring a Specific IP Address for Proxied Requests” in the *NetIQ Access Manager 4.2 Best Practices Guide*.
- Disabling Phishing For information about how to configure it, see “Disabling Phishing” in the *NetIQ Access Manager 4.2 Best Practices Guide*.

Modifying server.xml

You can modify the `server.xml` file to perform the following configurations:

- Customizing Certificate Errors. For information about how to configure it, see “Customizing Certificate Errors” on page 320.
- Configuring X.509 Authentication to Provide Access Manager Error Message. For information about how to configure it, see “Configuring X.509 Authentication to Provide Access Manager Error Message” on page 320.
- Securing the Embedded Service Provider Session Cookie on the Access Gateway. For information about how to configure it, see “Securing the Embedded Service Provider Session Cookie on the Access Gateway” on page 791.
- Configuring the SSL Communication. For information about how to configure it, see “Configuring the SSL Communication” on page 829.
- Protecting the Administration Console. For information about how to configure it, see “Protecting the Administration Console” in the *NetIQ Access Manager 4.2 Best Practices Guide*.Configuring the 256-bit and Higher Ciphers for SSL Communication. For information about how to configure it, see “Configuring the 256-bit and Higher Ciphers for SSL Communication ” in the *NetIQ Access Manager 4.2 Best Practices Guide*.Specifying SSL Configuration for Identity Server. For information about how to configure it, see “Configuring a Specific IP Address for Proxied Requests” in the *NetIQ Access Manager 4.2 Best Practices Guide*.

5 Configuring Authentication

The Identity Server is responsible for authenticating users, building the user's role information, and distributing it to various components. It also serves as the central point for components that request identity information.

This part discusses how to configure various types of authentications. Topics include:

- "Configuring Local Authentication" on page 271
- "Configuring Federated Authentication" on page 374

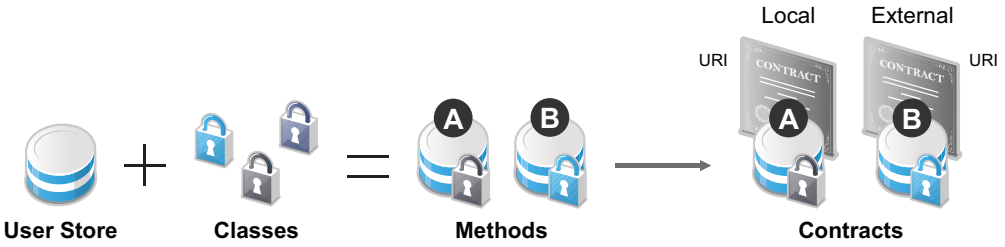
You can configure third-party authentication classes. You can also write your own Java class for authentication. For information about how to write your own class, see *NetIQ Access Manager 4.2 Developer Guide* and NetIQ Access Manager Developer Tools and Examples (https://www.netiq.com/documentation/access-manager-developer-documentation/).

Configuring Local Authentication

To guard against unauthorized access, Access Manager Appliance supports a number of ways for users to authenticate. These include name/password, RADIUS token-based authentication, and X.509 digital certificates. You configure authentication at the Identity Server by creating authentication contracts that Access Manager components (such as an Access Gateway) can use to protect a resource.

Figure 5-1 illustrates the components of a contract:

Figure 5-1 *Local Authentication*



- **User stores:** The user directories to which users authenticate on the back end. You set up your user store when you create an Identity Server cluster configuration. See "Configuring Identity User Stores" on page 272.
- **Classes:** The code (a Java class) that implements a particular authentication type (name/password, RADIUS, and X.509) or means of obtaining credentials. Classes specify how the Identity Server requests authentication information, and what it must do to validate those credentials. See "Creating Authentication Classes" on page 283.
- **Methods:** The pairing of an authentication class with one or more user stores, and whether the method identifies a user. See "Configuring Authentication Methods" on page 287.

- **Contracts:** The basic unit of authentication. Contracts can be local (executed at the server) or external (satisfied by another Identity Server). Contracts are identified by a unique URI that can be used by Access Gateways and agents to protect resources. Contracts are comprised of one or more authentication methods used to uniquely identify a user. You can associate multiple methods with one contract. See “Configuring Authentication Contracts” on page 289.

This section explains the following:

- “Configuring Identity User Stores” on page 272
- “Creating Authentication Classes” on page 283
- “Configuring Authentication Methods” on page 287
- “Configuring Authentication Contracts” on page 289
- “Specifying Authentication Defaults” on page 297
- “Social Authentication” on page 299
- “Two-Factor Authentication Using Time-Based One-Time Password (TOTP)” on page 305
- “Persistent Authentication” on page 308
- “RADIUS Authentication” on page 310
- “Client Integrity Check” on page 311
- “Mutual SSL (X.509) Authentication” on page 318
- “ORed Credential Class” on page 323
- “OpenID Authentication” on page 324
- “Password Retrieval” on page 325
- “Configuring Access Manager for NESCM” on page 327
- “Kerberos Authentication” on page 331
- “Risk-based Authentication” on page 343
- “Managing Direct Access to the Identity Server” on page 370

Configuring Identity User Stores

User stores are LDAP directory servers to which end users authenticate. You must specify an initial user store when creating an Identity Server configuration. You use the same procedure for setting up the initial user store, adding a user store, or modifying an existing user store.

1. In the Administration Console, click **Devices > Identity Servers** > **Servers** > **Edit** > **Local**.
2. Select from the following actions:

   New: To add a user store, click **New**. For configuration information, see “Configuring the User Store” on page 273.

   Delete: To delete a user store, select the user store, then click **Delete**.The user store list needs to contain at least one configured user store for the Identity Server to be functional.

   Modify: To modify the configuration of an existing user store, click the name of a user store. For configuration information, see “Configuring the User Store” on page 273.
3. Click **OK**, then update the Identity Server if you have modified the configuration.

See the following sections for specific configuration tasks:

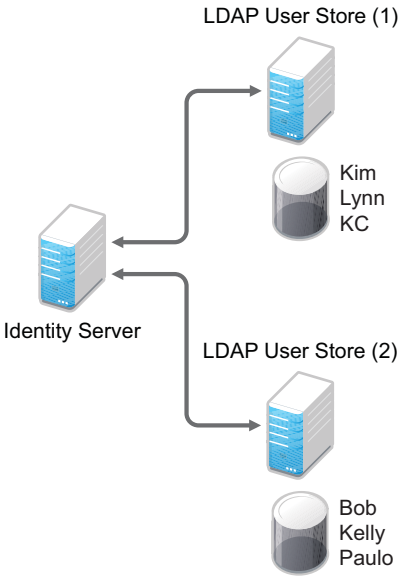
- “Using More Than One LDAP User Store” on page 273
- “Configuring the User Store” on page 273

- “Configuring an Admin User for the User Store” on page 276
- “Configuring a User Store for Secrets” on page 277

Using More Than One LDAP User Store

You can configure the Identity Server to search more than one user store during authentication. Figure 5-2 illustrates this type of configuration.

Figure 5-2 *Multiple LDAP Directories*



It is assumed that each LDAP directory contains different users. You must ensure that the users have unique names across all LDAP directories. If both directories contain a user with an identical name, the name and password information discovered in the search of the first directory is always used for authentication. You specify the search order when configuring the authentication method.

When users are added to the configuration store, objects are created for Access Manager profiles. If you delete a user from the LDAP directory, orphaned objects for that user remain in the configuration store.

If you add a secondary Administration Console and you have added replicas to the user store of the primary Administration Console, ensure that you also add the replicas to the secondary Administration Console.

All user stores that you add are included in health checks. If health problems are found, the system displays the user store on the Health page and in the trace log file.

Configuring the User Store

1. In the Administration Console, click **Devices > Identity Servers** > **Servers** > **Edit** > **Local**.
2. In the **User Stores** list, click **New** or the name of an existing user store.

   If you are creating an Identity Server configuration, this is Step 3 of the wizard.
3. Fill in the following fields:

   Name: The name of the user store for reference.

Admin Name: The distinguished name of the admin user of the LDAP directory, or a proxy user with specific LDAP rights to perform searches. For the LDAP extension to work, the proxy user requires write rights on the ACL. Administrator-level rights are required for setting up a user store. This ensures read/write access to all objects used by Access Manager. For more information about this user, see "Configuring an Admin User for the User Store" on page 276.

Each directory type uses a slightly different format for the DN:

- **eDirectory:** cn=admin,ou=users,o=novell
- **Active Directory:** cn=Administrator,cn=users,dc=domeh,dc=test,dc=com

  or cn=john smith,cn=users,dc=domeh,dc=test,dc=com
- **Sun ONE:** cn=admin,cn=users,dc=novell,dc=com

Admin Password and Confirm Password: Specify the password for the admin user and confirm it.

Directory Type: The type of LDAP directory. You can select **eDirectory**, **Active Directory**, or **Sun ONE**. If you have installed an LDAP server plug-in, you can select the custom type that you have configured it to use. For more information, see "LDAP Server Plug-In" in the NetIQ Access Manager 4.2 Developer Guide.

If eDirectory has been configured to use Domain Services for Windows, eDirectory behaves like Active Directory. When you configure such a directory to be a user store, its **Directory Type** must be set to Active Directory for proper operation.

Install NMAS SAML method: (eDirectory only) Extends the schema on the eDirectory server and installs an NMAS method. This method converts the Identity Server credentials to a form understood by eDirectory. This method is required if you have installed Novell SecretStore on the eDirectory server and you are going to use that SecretStore for Access Manager secrets. If you select this option, ensure that the admin you have configured for the user store has sufficient rights to extend the schema and add objects to the tree.

For additional configuration steps required to use secrets, see "Configuring a User Store for Secrets" on page 277.

Enable Secret Store lock checking: (eDirectory only) Enables Access Manager to prompt users for a passphrase when secrets are locked.

- If Access Manager is sharing secrets with other applications and these applications are using the security flag that locks secrets when a user's password is reset, you need to enable this option.
- If Access Manager is not sharing secrets with other applications, the secrets it is using are never locked, and you do not need enable this option.

4 Under **LDAP timeout settings**, specify the following:

LDAP Operation: Specify how long in seconds a transaction can take before timing out.

Idle Connection: Specify how long in seconds before connections begin closing. If a connection has been idle for this amount of time, the system creates another connection.

5 To specify a server replica, click **New**, then fill in the following fields:

For an eDirectory server, you must use a replica of the partition where the users reside. Ensure that each LDAP server in the cluster has a valid read/write replica. One option is to create a users partition (a partition that points to the OU containing the user accounts) and reference this server replica.

Name: The display name for the LDAP directory server. If your LDAP directory is replicated on multiple servers, use this name to identify a specific replica.

IP Address: The IP address of the LDAP directory server.

Port: The port of the LDAP directory server. Specify 389 for the clear text port, and 636 for the encrypted port.

Use secure LDAP connections: Specifies that the LDAP directory server requires secure (SSL) connections with the Identity Server.

This is the only configuration we recommend for the connection between the Identity Server and the LDAP server in a production environment. If you use port 389, usernames and passwords are sent in clear text on the wire.

This option must be enabled if you use this user store as a Novell SecretStore User Store Reference in the Credential Profile details. (See "Configuring Credential Profile Security and Display Settings" on page 475.) If you have specified that this user store is a SecretStore User Store Reference, this option is enabled but not editable.

NOTE: If the LDAP user store is using a trusted root which is not present in Trusted Roots or External Trusted Roots tab, you need to manually import the trusted root. For more information, see Chapter 14, "Managing Trusted Roots and Trust Stores," on page 815.

Connection limit: The maximum number of pooled simultaneous connections allowed to the replica. Valid values are between 5 and 50. How many you need depends upon the speed of your LDAP servers. If you modify the default value, monitor the change in performance. Larger numbers do not necessarily increase performance.

6 Select the replica, then click **Validate** to test the connection between the Identity Server and the replica.

The system displays the result under **Validation Status**. The system displays a green check mark if the connection is valid.

7 (Optional) To add additional replicas for the same user store, repeat Step 5 through Step 11.

Adding multiple replicas adds load balancing and failover to the user store. Replicas must be exact copies of each other.

For load balancing, a hash algorithm is used to map a user to a replica. All requests on behalf of that user are sent to that replica. Users are moved from their replica to another replica only when their replica is no longer available.

8 Add a search context.

The search context is used to locate users in the directory when a contract is executed.

- If a user exists outside of the specified search context (object, subtree, one level), the Identity Server cannot find the user, and the user cannot log in.
- If the search context is too broad, the Identity Server might find more than one match, in which case the contract fails, and the user cannot log in.

For example, if you allow users to have the same username and these users exist in the specified search context, these users cannot log in if you are using a simple username and password contract. The search for users matching this contract would return more than one match. In this case, you need to create a contract that specifies additional attributes so that the search returns only one match. For more information about how to create such contracts, see "Authentication Classes and Duplicate Common Names" on page 989.

IMPORTANT: For Active Directory, do not set the search context at the root level and set the scope to Subtree. This setting can cause serious performance problems. It is recommended that you set multiple search contexts, one for each top-level organizational unit.

---

9 Click **Finish**.

10 If prompted to restart Tomcat, click **OK**. Otherwise, update the Identity Server.

Configuring an Admin User for the User Store

The Identity Server must log in to each configured user store. It searches for users, and when a user is found, it reads the user's attributes values. When you configure a user store, you must supply the distinguished name of the user you want the Identity Server to use for logging in. You can use the admin user of your user store, or you can create a specialized admin user for the this purpose. When creating this admin user, you need to grant the following rights:

- The admin user needs rights to browse the tree, so the Identity Server can find the user who is trying to authenticate. The admin user needs browse rights to object class that defines the users and read and compare rights to the attributes of that class. When looking for the user, the Identity Server uses the GUID and naming attributes of the user class.

Directory	Object Class	GUID Attribute	Naming Attribute
eDirectory	User	guid	cn
Active Directory	User	objectGUID	sAMAccountName
Sun ONE	inetOrgPerson	nsuniqueid	uid

- The admin user needs read rights to any attributes used in policies (Role, Form Fill, Identity Injection,).
- If a secret store is used in Form Fill policies, the admin user needs write rights to the attributes storing the secrets.
- If a password management servlet is enabled, the admin user needs read rights to the attributes controlling grace login limits and remaining grace logins.
- If you use an LDAP extension, the user must have write rights on ACL allowing the user to check for account locks, password expiration, grace logins used, and so on.
- If you enable provisioning with the SAML or Liberty protocols, the admin user needs write rights to create users in the user store.
- If you use X.509 authentication, the admin user needs write rights to update the user's login status attributes.

If your user store is an eDirectory user store, Access Manager verifies that the admin user has sufficient rights to browse for users in the specified search contexts.

IMPORTANT: This check is not performed for Active Directory or Sun ONE. If your users cannot log in, you need to verify that you have given the admin for the user store sufficient rights to the specified search contexts.

Configuring a User Store for Secrets

Access Manager allows you to securely store user secrets. Secrets are a way to capture user input like Login ID and password credentials. These input data can later be reused or injected using Form Fill and Identity Injection policies. This feature is especially helpful when your Access Manager Credential Profile does not contain credentials for an application protected by Access Manager yet a single sign-on experience is required. Where and how the secrets can be stored is configurable and depends upon your user store:

- "Configuring the Configuration Datastore to Store Secrets" on page 277.

If you want to do minimal configuration, you can use the configuration datastore on the Administration Console to store the secrets. This option can be used without changes, but is recommended only for use in small Access Manager environments. To increase the security of the secrets, NetIQ recommends that you change the default security options.

IMPORTANT: Using this option will put additional load on your Administration Console and introduces login delays compared to other options. Therefore it is recommended that this option is used wisely.

- "Configuring an LDAP Directory to Store the Secrets" on page 278.

  This is the recommended option and can be used with any LDAP directory. To use this option, extend the schema to add an attribute to your user object on the LDAP directory that will encrypt and store the secrets.

- "Configuring an eDirectory User Store to Use SecretStore" on page 279.

  If your user store is eDirectory and you have installed Novell SecretStore, you can choose to use the SecretStore on your eDirectory server to store the secrets. This differs from the schema extension method as Novell SecretStore can also be accessed and managed by Novell SecureLogin. This allows secrets to be shared with SecureLogin to provide a thick client single sign-on while Access Manager can provide a web single sign-on experience without credential collisions.

For troubleshooting tips, see "Troubleshooting the Storing of Secrets" on page 281.

Configuring the Configuration Datastore to Store Secrets

When you use the configuration datastore of the Administration Console as the secret store, the nidswsfss attribute of the nidsLibertyUserProfile object is used to store the secrets.

1. In the Administration Console, click **Devices > Identity Servers** > *Edit* > **Liberty > Web Service Provider**.
2. Click **Credential Profile**.
3. Scroll to the **Local Storage of Secrets** section and configure the following security options:

   Encryption Password Hash Key: (Required) Specify the password that you want to use as a seed to create the encryption algorithm. To increase the security of the secrets, we recommend that you change the default password to a unique alphanumeric value.

   IMPORTANT: Before using Access Manager to store and encrypt secrets, ensure that you choose your **Preferred Encryption Method** and change the default **Encryption Password Hash Key** value. If either of these options are changed after any secrets are stored, Access Manager will not be able to retrieve the secrets.

   Preferred Encryption Method: Specify the preferred encryption method. Select the method that complies with your security model:

   - **Password Based Encryption With MD5 and DES:** MD5 is an algorithm that is used to verify data integrity. Data Encryption Standard (DES) is a widely used method of data encryption that uses a private key.
   - **DES:** Data Encryption Standard (DES) is a widely used method of data encryption that uses a private key. Like other private key cryptographic methods, both the sender and the receiver must know and use the same private key.
   - **Triple DES:** A variant of DES in which data is encrypted three times with standard DES, using two different keys.

Extended Schema User Store References: Do not specify a user store reference. When this option contains no values, the configuration datastore is used to store the secrets.

4 Click **OK**.

5 On the Identity Servers page, update the Identity Server.

6 To use the secret store to store policy secrets, see "Creating and Managing Shared Secrets" on page 732.

Configuring an LDAP Directory to Store the Secrets

When you use an LDAP directory to store the secrets, you need to enable the user store for the secrets. You select the LDAP directory, then specify an attribute. The attribute you specify is used to store an XML document that contains encrypted secret values. This attribute must be a single-valued case ignore string that you have defined and assigned to the user object in the schema.

To use an LDAP directory to store secrets, your network environment must conform to the following requirements:

- The user class object must contain an attribute that can be used to store the secrets. This attribute must be a string attribute that is single valued and case ignore.
- The user store must be configured to use secure connections (click **Devices** > **Identity Servers** > **Edit** > **Local** > **User Stores** > **[User Store Name]**. In the **Server replicas** section, ensure that the **Port** is 636 and that **Use SSL** is enabled. If they aren't, click the name of the replica and reconfigure it.

To configure the LDAP directory:

1 In the Administration Console, click **Devices > Identity Servers** > *Edit* > **Liberty > Web Service Providers**.

2 Click **Credential Profile**.

3 Scroll to the **Local Storage of Secrets** section and configure the following options:

Encryption Password Hash Key: (Required) Specifies the password that you want to use as a seed to create the encryption algorithm. To increase the security of the secrets, we recommend that you change the default password to a unique alphanumeric value.

Preferred Encryption Method: Specifies the preferred encryption method. Select the method that complies with your security model:

- **Password Based Encryption With MD5 and DES:** MD5 is an algorithm that is used to verify data integrity. Data Encryption Standard (DES) is a widely used method of data encryption that uses a private key.
- **DES:** Data Encryption Standard (DES) is a widely used method of data encryption that uses a private key. Like other private key cryptographic methods, both the sender and the receiver must know and use the same private key.
- **Triple DES:** A variant of DES in which data is encrypted three times with standard DES, using two different keys.

IMPORTANT: Before using Access Manager to store and encrypt secrets, ensure that you choose your **Preferred Encryption Method** and change the default **Encryption Password Hash Key** value. If either of these options are changed after any secrets are stored, Access Manager will not be able to retrieve the secrets.

---

4 To specify where to store secret data, click **New** under **Extended Schema User Store References** and fill in the following:

User Store: Select the user store where you want secret store enabled.

Attribute Name: Specify the LDAP attribute that you have created to store the secrets on the selected user store.

5 Click **OK** twice.

6 On the Identity Servers page, update the Identity Server.

7 To create policies that use the stored secrets, see "Creating and Managing Shared Secrets" on page 732.

For troubleshooting information, see "Troubleshooting the Storing of Secrets" on page 281.

Configuring an eDirectory User Store to Use SecretStore

For Access Manager to use Novell SecretStore, the user store must be eDirectory and Novell SecretStore must be installed there. When configuring this user store for secrets, Access Manager extends the eDirectory schema for an NMAS method. This method converts authentication credentials to a form understood by eDirectory. For example, Access Manager supports smart card and token authentications, and these authentication credentials must be converted into the username and password credentials that eDirectory requires. This allows the Identity Server to authenticate as that user and access the user's secrets. Without this NMAS method, the Identity Server is denied access to the user's secrets.

To use a remote SecretStore, your network environment must conform to the following requirements:

- The eDirectory server must have Novell SecretStore installed.
- When you configure a user store to use Novell SecretStore, the admin user that you have configured for the user store must have sufficient rights to extend the schema on the eDirectory server, to install the SAML NMAS method, and set up the required certificates and objects. For more information about the rights required, see "Configuring an Admin User for the User Store" on page 276.
- The user store must be configured to use secure connections (click **Access Manager** > **Identity Servers** > **Edit** > **Local** > **User Stores** > **[User Store Name]**. In the **Server replicas** section, ensure that the **Port** is 636 and that **Use SSL** is enabled. If they aren't, click the name of the replica and reconfigure it.

  NOTE: While configuring new replicas for the same user store, by default the **Use secure LDAP connections** option will be selected and the default port will be 636. The **Use secure LDAP connections** option will be non-editable.

- If you have enabled a firewall between the Administration Console and the user store, and between the Identity Server and the user store, ensure that both LDAP ports (389 and 636) and the NCP port (524) are opened.
- If you are going to configure Access Manager to use secrets that are used by other applications, you need to plan a configuration that allows the user to unlock a locked SecretStore. See "Determining a Strategy for Unlocking the SecretStore" on page 280.

To configure the user store:

1 In the Administration Console, click **Devices > Identity Servers** > *Edit* > *Local*.

2 Click the name of your user store.

3 Select **Install NMAS SAML method**, then click **OK**.

This installs a required NMAS method in the eDirectory schema and adds required objects to the tree.

IMPORTANT: If your eDirectory user store is running on SLES 11 SP1 64-bit operating system (or a higher version), the eDirectory server is missing some support libraries that this SAML method requires. For information about installing these libraries, see TID 7006437 (http://www.novell.com/support/viewContent.do?externalId=7006437&sliceId=1).

4 Click **Liberty > Web Service Providers**.

5 Click **Credential Profile**.

6 Scroll to the **Remote Storage of Secrets** section.

7 Click **New** under **Novell Secret Store User Store References**.

This adds a reference to a user store where SecretStore has been installed.

8 Click the user store that you configured for SecretStore.

9 Click **OK** twice.

10 On the Identity Servers page, update the Identity Server.

11 Continue with one of the following:

- If other applications are using the secret store, you need to determine whether Access Manager users need the option to unlock the secret store. See "Determining a Strategy for Unlocking the SecretStore" on page 280.
- To create policies that use the stored secrets, see "Creating and Managing Shared Secrets" on page 732.
- For troubleshooting information, see "Troubleshooting the Storing of Secrets" on page 281.

Determining a Strategy for Unlocking the SecretStore

When an administrator resets a user's password, secrets written to the Novell SecretStore with an enhanced security flag become locked. The Identity Server does not write the secrets that it creates with this flag, but other applications might:

- If Access Manager is not sharing secrets with other applications, the secrets it is using are never locked, and you do not need to configure Access Manager to unlock secrets.
- If Access Manager is sharing secrets with other applications and these application are using the security flag that locks secrets when a user's password is reset, you need to configure Access Manager so that users can unlock their secrets.

If you want users to receive a prompt for a passphrase when secrets are locked, complete the following configuration steps:

1 Require all users to set up a passphrase (also called the Master Password).

Access Manager uses the SecretStore Master Password as the passphrase to unlock the secrets. If the user has not set a passphrase before the SecretStore is locked, this feature of Access Manager cannot unlock the SecretStore. If it is necessary to unlock the SecretStore by using the user's prior password, another tool must be used. See your SecretStore documentation.

2 Configure the Identity Server to perform the check:

2a In the Administration Console, click **Devices** > **Identity Servers** > **Edit** > **Local** > **[User Store Name]**.

2b Select the **Enable Secret Store lock checking** option.

2c Click **OK** twice, then update the Identity Server.

3 Ensure that Web Services Framework is enabled:

3a In the Administration Console, click **Devices** > **Identity Servers** > **Edit** > **Liberty** > **Web Services Framework**.

3b In the **Framework General Settings** section, ensure that **Enable Framework** is selected.

3c Click **OK**. If you made any changes, update the Identity Server.

4 Continue with "Creating and Managing Shared Secrets" on page 732.

When the SecretStore is locked and the users log in, the users are first prompted for their login credentials, then prompted for the passphrase that is used to unlock the SecretStore.

Troubleshooting the Storing of Secrets

- "Secrets Are Not Stored in Novell SecretStore" on page 281
- "Users Are Receiving Invalid Credential Messages" on page 283
- "Secrets Aren't Stored in the LDAP Directory" on page 283

Secrets Are Not Stored in Novell SecretStore

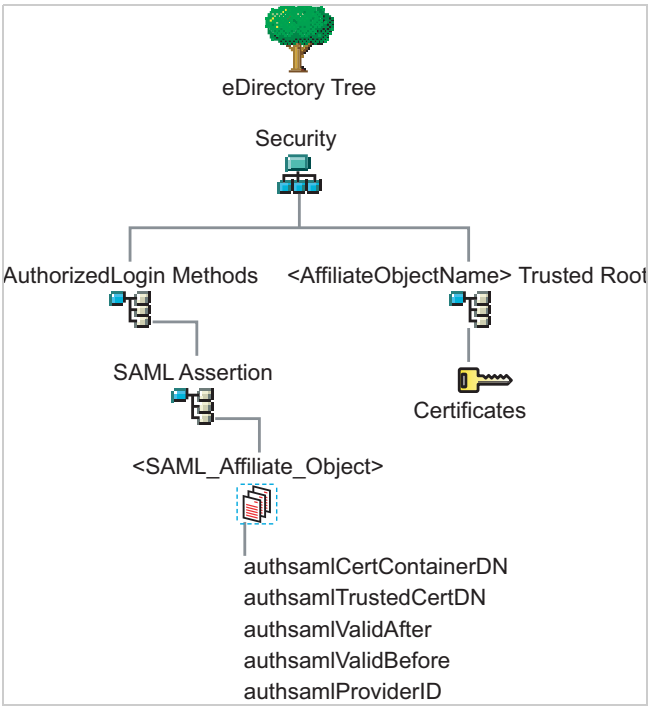
When you use Novell SecretStore to store the secrets, the schema on the eDirectory server must be extended, and specific SAML objects and certificates must be created.

To verify that the schema was extended and the objects were created on the eDirectory server:

1 Open an LDAP browser and connect to the LDAP server.

2 Browse to the Security container.

3 Look for objects similar to the following:



If the schema has been extended correctly, you can find a SAML Assertion object in the Authorized Login Methods container. The SAML_Assertion object contains an alphanumeric generated name for a SAML affiliate object. This object has four attributes.

The SAML affiliate object name is used to generate another container in the Security container. This new container is the <*AffiliateObjectName*> Trusted Root container that contains public key signing certificate.

4 Complete one of the following:

- If these objects do not exist, verify the following, then continue with Step 5:
  - The admin user for the user store has sufficient rights to extend the schema and add these objects to the Security container.
  - Any configured firewalls must allow NCP and LDAP traffic for the Administration Console, the Identity Server, and the LDAP user store.
- If the objects exist, check for time synchronization problems. For more information, see "Users Are Receiving Invalid Credential Messages" on page 283.

5 In the Administration Console, modify the secret store configuration so that it is resent to the user store:

5a Click **Devices > Identity Servers** > *Edit* > **Liberty > Web Service Providers** > **Credential Profile**.

5b In the **Remote Storage of Secrets** section, remove the user store, then add it again.

5c Click **OK**.

6 On the Identity Servers page, update the Identity Server.

Users Are Receiving Invalid Credential Messages

The <SAML_Affiliate_Object>.SAML-Assertion.AuthorizedLoginMethods.Security object contains two attributes that determine how long credentials are valid. If your Identity Server and eDirectory server are not time synchronized, the credentials can become invalid before a user has time to use them.

Either ensure that the time of your Identity Server and eDirectory server are synchronized, or increase the value of the authsamlValidAfter and authsamlValidBefore attributes of the SAML affiliate object.

Secrets Aren't Stored in the LDAP Directory

1 Open an LDAP browser and connect to the eDirectory server.

2 Browse to the user object.

3 Verify that the user object contains the LDAP attribute that you have specified as the attribute to store the secrets.

4 If the attribute exists, browse to the schema and verify that the attribute has the following characteristics:

- Single valued
- Case ignore
- String

Creating Authentication Classes

Authentication classes let you define ways of obtaining end user credentials.You specify the code (Java class) and properties to be executed to implement a particular authentication type.

Several authentication classes are included with Access Manager to provide a variety of ways to authenticate end users. Custom authentication classes provided by other vendors can also be configured to run in the system.

1 In the Administration Console, click **Devices > Identity Server > Edit > Local > Classes**.

The following classes are predefined for Access Manager:

Introductions: When the class is configured, it allows users to select an identity provider from a list of introducable identity providers. For information about how to configure and use this class, see “Configuring the Introductions Class” on page 148.

Name/Password - Basic: Basic authentication over HTTP using a standard login pop-up page provided by the Web browser.

Name/Password - Form: Form-based authentication over HTTP or HTTPS.

Secure Name/Password - Basic: Basic authentication over HTTPS using a standard login page provided by the Web browser.

Secure Name/Password - Form: Form-based authentication over HTTPS.

Trust Levels: When this class is configured, it defines authentication levels for classes that can be used in authentication requests. For more information about how to configure and use this class, see “Configuring the Trust Levels Class” on page 149.

2 To delete a class, select the class, then click **Delete**.

You cannot delete a class if a method is using it.

For information about how to create a name/password class, see the following sections:

- “Creating Basic or Form-Based Authentication Classes” on page 284.
- “Specifying Common Class Properties” on page 285

Some classes require additional configuration to enable their use for authentication. See the following sections:

- “RADIUS Authentication” on page 310
- “Mutual SSL (X.509) Authentication” on page 318
- “ORed Credential Class” on page 323
- “OpenID Authentication” on page 324
- “Password Retrieval” on page 325
- “Configuring Access Manager for NESCM” on page 327
- “Kerberos Authentication” on page 331
- “Two-Factor Authentication Using Time-Based One-Time Password (TOTP)” on page 305
- “Risk-based Authentication” on page 343

Creating Basic or Form-Based Authentication Classes

1 In the Administration Console, click **Devices > Identity Server > Edit > Local > Classes**.

2 Click **New** to launch the **Create Authentication Class Wizard**.

3 Specify a display name, then select a class from the **Java class** drop-down menu.

- The following classes are recommended only for testing purposes:

  BasicClass: Uses basic HTTP authentication.

  PasswordClass: Passes the user name and password over HTTP in readable text, and uses a form-based login to collect the name and password.

  RadiusClass: RADIUS enables communication between remote access servers and a central server. For a production environment, use ProtectedRadiusClass.

- For a production environment, select one of the following protected classes:

  X509Class: Certificate-based authentication. See "Mutual SSL (X.509) Authentication" on page 318.

  SocialAuthClass: The authentication class used for implementing authentication through external OAuth providers such as Facebook, GooglePlus, LinkedIn and Twitter. See "Social Authentication" on page 299.

  TOTPClass: The authentication class used for implementing two-factor authentication. See "Two-Factor Authentication Using Time-Based One-Time Password (TOTP)" on page 305.

  Risk-based Auth Class: The authentication class used for assessing the risk after authentication. See "Risk-based Authentication" on page 343.

  Risk-based Pro-Auth Class: The authentication class used for assessing the risk before authentication. See "Risk-based Authentication" on page 343.

  ProtectedBasicClass: BasicClass protected by HTTPS.

  ProtectedPasswordClass: PasswordClass protected by HTTPS (form-based).

  ProtectedRadiusClass: RadiusClass protected by HTTPS. See "RADIUS Authentication" on page 310 for configuration steps.

  KerberosClass: The authentication class used for using Kerberos for Active Directory and Identity Server authentication. See "Kerberos Authentication" on page 331 for configuration steps.

  NMASAuthClass: The authentication class used for Novell Modular Authentication Services (NMAS), which uses fingerprint and other technology as a means to authenticate a user. For instructions on using the NMAS NESCM method, see "Configuring Access Manager for NESCM" on page 327.

  NPOrRadiusOrX509Class: The authentication class that allows the creation of a contract from which the user can select an authentication method: name/password, RADIUS, or X.509. For configuration information, see "ORed Credential Class" on page 323.

  PasswordFetchClass: The authentication class that allows the Identity Server to retrieve the user's password when the user has used a non-password class for authentication. For configuration information, see "Password Retrieval" on page 325.

  PersistentAuthClass: The authentication class that allows for persistent logins, long authentication sessions, or remember my password functionality. For configuration information, see "Persistent Authentication" on page 308.

  Other: Used for third-party authentication classes or if you have written your own Java class. For information about how to write your own class, see Novell Access Manager Developer Tools and Examples (https://www.netiq.com/documentation/access-manager-developer-documentation/).

  AliasUserPasswordClass: This class supports authentication of a user against user's alias name. This class uses the alias object of the user object and the password of the corresponding user object to authenticate.

4 Click **Next** to configure the properties for each class. Click **New**, then enter a name and value. The names and values you enter are case sensitive. See "Specifying Common Class Properties" on page 285 for the properties that are used by the basic and password classes.

5 Click **Finish**.

6 Continue with "Configuring Authentication Methods" on page 287.

To use an authentication class, the class must have one or more associated methods.

Specifying Common Class Properties

The following properties can be used by the basic and password classes:

- "Query Property" on page 286
- "JSP Property" on page 286
- "MainJSP Property" on page 287

These properties can also be specified on a method derived from the class. If you are going to create multiple methods from the same class, consider the following conditions:

- If you want the methods to share the same properties, you can save configuration steps by defining the properties on the class.
- If you want the methods to use different values for the properties such as one method specifying one custom login page and another method specifying a different custom login page, then you must specify the properties on the method.

Query Property

Normally, the Identity Server uses the username to find a user in the user store. You can change this behavior by using the Query property. This property determines the username value for authentication. The default Query string prompts the users for the value of the CN attribute. You can modify this by requesting a different attribute in the LDAP query.

The Query property can be used by the following classes:

- BasicClass
- PasswordClass
- ProtectedBasicClass
- ProtectedPasswordClass

For example, to query for the user's UID attribute to use for the username, you would specify the following query:

Property Name: Query

Property Value: `(&(objectclass=person)(uid=%Ecom_User_ID%))`

The values are case sensitive. The name of the property must be Query with an initial capital. The `%Ecom_User_ID%` variable is used in the default `login.jsp` for the username in the four classes that support the Query property. The variable is replaced with the value the user enters for his or her username, and the LDAP query is sent to the user store to see if the user's attribute value matches the entered value. You can specify any attribute for the Query that is defined in your user store for the object class of person and that is used to identify the user.

The Query you define for the BasicClass and the ProtectedBasicClass needs to use an attribute that your users define as their username. The PasswordClass and the ProtectedPasswordClass do not have this requirement. They also support the JSP property, which allows you to specify a custom login.jsp and have it prompt for other attributes that can be used for login.

For example, you can define the following Query to prompt the users for their email address rather than their username.

Property Name: Query

Property Value: `(&(objectclass=person)(email=%EMail Value%))`

The `%EMail Value%` must match the variable in the custom login page that is filled in when the users enter their credentials. The `objectclass` value must be a valid object class in the LDAP user store. The email attribute must be a valid attribute of the person class.

When you specify such a Query, you must also modify the login page to prompt the user for the correct information. Instead of prompting the user for a username, the login form must prompt the user for an e-mail address. The JSP Property allows you to specify a custom login page. For information about creating a custom login page, see “Customizing the Identity Server Login Page” on page 193.

JSP Property

The JSP property allows you to specify a custom login page. This property can be used with the following classes:

- PasswordClass
- ProtectedPasswordClass

The property name is JSP and the property value is the filename of the login page you customized without the `.jsp` extension of the file. The property value cannot contain `nidp` in its name.

For example, if you created a custom file named `emaillogin.jsp`, you would specify the following values. The values are case sensitive. The property name needs to be entered as all capitals.

Property Name: JSP

Property Value: `emaillogin`

If you use two methods to create a contract, this property must be set to the same value on both or set on only one. When it is set on only one method, the value is applied to both. This property needs to be used with the MainJSP Property. For information about how to create a custom login page, see “Customizing the Identity Server Login Page” on page 193.

MainJSP Property

When the MainJSP property is set to true, it indicates that you want to use the page specified in the JSP property for the login page. When this property is set to false, which is the default value, the `nidp.jsp` is used for the login page. If you use two methods to create a contract, this property must be set to the same value on both or set on only one. When it is set on only one method, the value is applied to both.

Property Name: `MainJSP`

Property Value: `true`

For information about how to create a custom login page, see “Customizing the Identity Server Login Page” on page 193.

Configuring Authentication Methods

Authentication methods let you associate authentication classes with user stores. You use a particular authentication class to obtain credentials about an entity, and then validate those credentials against a list of user stores.

After the system locates the entity in a particular user store, no further checking occurs, even if the credentials fail to validate the entity. Typically, the entity being authenticated is a user, and the definition of an authentication method specifies whether this is the case. You can alter the behavior of an authentication class by specifying properties (name/value pairs) that override those of the authentication class.

To configure a method for an authentication class:

1 In the Administration Console, click **Devices > Identity Servers > Edit > Local > Methods.**

2 To delete a method, select the method, then click **Delete**.

A method cannot be deleted if a contract is using it.

3 To modify an authentication method, click its name, or to create one, click **New**.

4 Fill in the following fields:

Display Name: The name to be used to refer to the new method.

Class: The authentication class to use for this method. See “Creating Authentication Classes” on page 283.

Identifies User: Specifies whether this authentication method must be used to identify the user. Usually, you must enable this option. When configuring multiple methods for a contract, you might need to disable this option for some methods.

If you enable this option on two or more methods in a contract, these methods need to identify the same user in the same user store.

If you enable this option on just one method in the contract, that method identifies the user when the authentication method succeeds. The other methods in the contract must succeed, but might not authenticate the user. For example, the method that identifies the user could require a name and a password for authentication, and the other method in the contract could prompt for a certificate that identifies the user’s computer.

To achieve SSO on back end web application when the passwordfetch class is enabled, see TID.

NOTE: If the policy is using `Risk-based Pre-Auth Class`, this options must be selected at least in one method of the contract.

Overwrite Temporary User: If you select this check box, then the temporary user credentials profile got form previous authentication method in the same session will be overwritten with real user credentials profile got from this authentication method.

Overwrite Real User: If you select this check box, then the real user credentials profile got form previous authentication method in the same session will be overwritten with real user credentials profile got from this authentication method.

5 Add user stores to search.

You can select from the list of all the user stores you have set up. If you have several user stores, the system searches through them based on the order specified here. If a user store is not moved to the **User stores** list, users in that user store cannot use this method for authentication.

<Default User Store>: The default user store in your system. See “Specifying Authentication Defaults” on page 297.

6 (Optional) Under Properties, click **New**, then fill in the following fields:

Property Name: The name of the property to be set. This value is case sensitive and specific to an authentication class. The same properties that can be set on an authentication class can be set on the method.

You can use the method properties to override the property settings specified on the authentication class. For example, you might want to use the authentication class for multiple companies, but use a slightly different login page that is customized with the company's logo. You can use the same authentication class, create a different method for each company, and use the JSP property to specify the appropriate login page for each company.

For information about the available properties for the basic and form classes, see "Specifying Common Class Properties" on page 285

The Radius classes have the following additional properties that can be set on the method:

- **RADIUS_LOOKUP_ATTR:** Defines an LDAP attribute whose value is read and used as the ID is passed to the RADIUS server. If not specified, the user name entered is used.
- **NAS_IP_ADDRESS:** Specifies an IP address used as a RADIUS attribute. You might use this property for situations in which service providers are using a cluster of small network access servers (NASs). The value you enter is sent to the RADIUS server.

If this method is part of a multi-factor authentication, you can set the following additional property:

- **PRINCIPAL_MISMATCH_ERR:** Specifies the error message to be displayed if this method identifies a different principal than other methods in the multi-factor authentication.

Property Value: The values associated with the **Property Name** field.

7 Click **Finish**.

8 Continue with "Configuring Authentication Contracts" on page 289.

To use a method for authenticating a user, each method must have an associated contract.

Configuring Authentication Contracts

Authentication contracts define how authentication occurs. An Identity Server can have several authentication contracts available, such as name/password, X.509, or Kerberos. From the available contracts, you assign a contract to a specific resource or resources. It is access to a resource that triggers the authentication process. If the user has already supplied the required credentials for the contract, the user is not prompted for them again.

Each contract is assigned a URI that uniquely identifies it. This URI can be shared with other providers so that they can identify the type of credentials the Identity Provider is requiring. You can also restrict a contract so that it can only be used for local authentication and not with other providers.

1 In the Administration Console, click **Devices > Identity Servers > Edit > Local > Contracts**.

2 To delete a contract, select the contract, then click **Delete**.

You cannot delete a contract if it is in use by an Access Gateway.

3 To create a new contract, click **New**.

4 Fill in the following fields:

Display name: Specifies the name of the authentication contract.

URI: Specifies a value that uniquely identifies the contract from all other contracts. It is used to identify this contract for external providers and is a unique path value that you create. No spaces can exist in the URI field.

The following are all valid values for the URI:

```
/mycompany/name/password/form
http://mycompany.com/login
secure/form/password/bcompany
```

Password expiration servlet: Specifies a URL to a page where the user can change password when the password expires or is within the grace login period. You must use eDirectory to change the number of grace logins. Grace logins work only with eDirectory.

For more information about how to use this type of servlet, see “Using a Password Expiration Service” on page 293.

Allow User Interaction: If you specify a password expiration servlet, you can enable this option, which allows the users to decide whether to go to the servlet and change their passwords or to skip the servlet. If you always want to force the users to go the servlet to change their passwords, do not enable this option.

Login Redirect URL: You will be redirected to the URL specified in this field. Use this field for the following scenarios:

- Forcing the user to a specific home page after successful Access Manager authentication.
- Forcing the user to configure challenge/response forgotten password questions

For more information about the URL parameters, see “Using Login Redirect URL Parameters” on page 296.

Allow User Interaction: You can enable this option, which allows the user to decide whether to continue to access a pre-configured URL or to continue to the page that the user usually accesses. For example, the user may usually access www.a.com and have specified the redirect URL as `https://someservice.com/path/password?user=<USERID>&store=<STOREID>&returl=<RETURN_URL>` then, continue will allow you to continue with the website you access i.e. `www.a.com` and redirect URL will take you to the URL `https://someservice.com/path/password?user=<USERID>&store=<STOREID>&returl=<RETURN_URL>&action=expire` and then to `www.a.com`.

Authentication Level: A number you can assign to this authentication contract to specify its security level or rank. You use this setting to preserve authentication contracts of a higher security level. When you enable the **Satisfiable by a contract of equal or higher level** option on this page, the system uses this value as a reference.

For example, you might create a name/password authentication contract and assign it to level one. You might also create an X.509 authentication contract and assign it to level two. If a user supplies the credentials for the X.509 level-two contract, the system does not require the credentials to satisfy the name/password level-one authentication contract.

Authentication Timeout: Specify how long the session can be inactive before the user is prompted to log in again. The value can be from 5 minutes to 66535 minutes and must be divisible by 5.

If you modify the timeout value for a contract, the newly assigned value is given to users as they log in. Currently logged in users retain the old value until they re-authenticate.

You need to experiment to discover what values are best for your network configuration, your security requirements, and your users.

- Shorter timeouts increase back-channel traffic and require more threads for authentication checks, but quickly free resources that are being used by inactive users. If you have slow back-end services, users could get disconnected waiting for a response, and these disconnects can generate more authentication traffic.
- Longer timeouts, which allow inactive users to remain connected, increase memory requirements to store session information, but require fewer threads and don't generate as much back-channel traffic.

For example, if you set the timeout to 5 minutes, an authentication check needs to be done 12 times each hour for each user authenticating with this contract. If the timeout is set to 60 minutes, an authentication check is done only one time each hour for each user. However, for the 5 minute timeout, resources can be freed within 5 minutes of inactivity by the user. For the 60-minute timeout, resources can take as long as 60 minutes to be freed, depending upon when the user goes inactive.

NOTE: In case of **Name/Password - Basic** and **Secure Name/Password - Basic** contracts applied to a protected resource, then you won't find the session as timed out, as the session gets renewed after timeout without user intervention using the Basic header sent from browser to Identity Provider.

For information about how to use this feature with the Access Gateway, see "Assigning a Timeout Per Protected Resource" on page 110.

Activity Realm(s): Specify the name of the realm that can be used to indicate activity. Use a comma-separated list to specify multiple realms. This allows a user's session to be kept alive when the user is accessing resources that are protected by different contracts. If both contracts belong to the same realm, activity on either resource keeps the session alive on the other resource. For more information about this feature, see "Using Activity Realms" on page 296.

Satisfiable by a contract of equal or higher level: Allows the system to satisfy this authentication contract if a user has logged in using another contract of an equal or higher authentication level, as specified in the **Authentication Level** field of an authentication contract.

When you enable this option, you need to be aware of the authentication levels you have set for other contracts and the level that has been assigned to the default contract.

When the protected resource is configured with **Name/Password -Form** as Authentication procedure, the user authentication details are prompted with transient federation. This option must be enabled to avoid prompting for authentication in the Target Service Provider.

Satisfiable by External Provider: Allows this contract to be selected when configuring an identity provider for Liberty or SAML 2.0. When you configure the authentication request, you can select a contract that has this option enabled and require the identity provider to use this contract in order for authentication to succeed.

Requested By: Select one of the following options:

- **Do not specify:** Specifies that the identity provider can send any type of authentication to satisfy a service provider's request, and instructs a service provider to not send a request for a specific authentication type or contract.
- **Use Types:** Specifies that authentication types must be used.

  Select the types from the **Available types** field to specify which type to use for authentication between trusted service providers and identity providers. Standard types include Name/Password, Secure Name/Password, X509, Token, and so on.
- **Use Contracts:** Specifies that authentication contracts must be used.

  Select the contract from the **Available contracts** list. For a contract to appear in the **Available contracts** list, the contract must have the **Satisfiable by External Provider** option enabled. To use the contract for federated authentication, the contract's URI must be the same on the identity provider and the service provider. For information about contract options, see "Configuring Authentication Contracts" on page 289.

  Most third-party identity providers do not use contracts.

Allowable Class: Specifies the class that instructs a service provider to send a request for a specific authentication type to the Identity Provider. You are allowed to modify this option only when you select authentication types.

NOTE: In SAML 2.0 federation with Access Manager as a service provider, if external identity provider is authenticating a user, it sends `<AuthnContext>` information after authentication in the response. Access Manager uses this `<AuthnContext>` to find a matching contract at the service provider to identify the user. It identifies the contract by trying to match `<saml:AuthnContextClassRef>` with AllowableClass attribute or `<saml:AuthnContextDeclRef>` with URI attribute of existing contracts at the service provider.

For example, if the external identity provider sends the following AuthnContext:

```
<saml:AuthnContext>
<saml:AuthnContextClassRef>urn:oasis:names:tc:SAML:2.0:ac:classes:Password</
saml:AuthnContextClassRef> <saml:AuthnContextDeclRef>adroit:login:user:np</
saml:AuthnContextDeclRef> </saml:AuthnContext>
```

and if Access Manager(as a Service Provider) has a contract A with `uri = adroit:login:user:np` or with Allowable class = `urn:oasis:names:tc:SAML:2.0:ac:classes:Password`, then it matches the contract.

NOTE: The Allowable class field is blank when an inbuilt Authentication Class is used in the Identity Server.

For more information about using CloudAccess as a trusted Identity Provider, see Using NetIQ® CloudAccess as a Trusted Identity Provider for NetIQ® Access Manager.

Methods and Available Methods: Specifies the authentication method to use for the contract. You can specify the order in which the methods are executed for login; however, this is not a graded list, so all the methods you specify are required. **Available methods** are the authentication methods you have set up.

You can enable the multi-factor authentication by associating more than one methods to a contract.

If you add more than one X.509 method, only the first one is used and it is automatically moved to the top of the list.

When you choose a secure method, such as Secure Name/Password, ensure that you have enabled security for the Identity Server configuration by setting the protocol to HTTPS. See Chapter 15, “Enabling SSL Communication,” on page 819.

5 Click **Next**.

6 Configure a card for the contract by filling in the following:

ID: (Optional) Specify an alphanumeric value that identifies the card. If you need to reference this card outside of the Administration Console, you need to specify a value here. If you do not assign a value, the Identity Server creates one for its internal use.

Text: Specify the text that is displayed on the card to the user.

Image: Specify the image to be displayed on the card. Select the image from the drop down list. To add an image to the list, click **Select local image**.

Show Card: Determine whether the card is shown to the user, which allows the user to select and use the card for authentication. If this option is not selected, the card is only used when a service provider makes a request for the card.

Passive Authentication Only: Select this option if you do not want the Identity Server to prompt the user for credentials. If the Identity Server can fulfill the authentication request without any user interaction, the authentication succeeds. Otherwise, it fails.

7 Click **Finish**, then **OK**.

8 Update the Identity Server and any devices that use the Identity Server configuration.

9 To use this contract, you must configure Access Manager to use it:

- You can assign it as the default contract for the Identity Server. See “Specifying Authentication Defaults” on page 297.
- You can configure a protected resource to use it. See Chapter , “Protecting Web Resources Through the Access Gateway,” on page 92.

Configuring Options for an Authentication Contract

You can configure an authentication contract to perform the following actions:

- To redirect a user trying to log in to the authentication contract with an expired password to the password management URI. For more information about how and where to use this option, see “Redirection to Password Management Servlet Protected by Access Gateway When Password Expires” on page 295.
- To hide contracts with equal levels.

Perform the following steps:

1 In the Administration Console, click **Identity Servers > Edit > Local > Contracts > [*Contract Name*]**.

2 Click **Options** > **New**.

3 Specify the following details:

Property	Description
AUTHENTICATE WITH EXPIRED PASSWORD	Select the **Property Value** as **true** if you want to redirect a user logging with an expired password to the password management URI protected by an Access Gateway.
HIDE CARDS WITH EQUAL LEVEL	Select the **Property Value** as **true** if you want to disable showing any other higher level authentication cards, if **Satisfiable by a contract of equal or higher level** is enabled.
OTHER	Specify **Property Name** and **Value** if you want to configure any other property for this contract.

4 Click **OK**.

Using a Password Expiration Service

Access Manager works with any password management service that works with your user store. For an implementation example, see Configuring Access Manager for UserApp and SAML.

Configure the following options:

- “URL Parameters” on page 293
- “Forcing Authentication after the Password Has Changed” on page 294
- “Grace Logins” on page 295
- “Federated Accounts” on page 295
- “Redirection to Password Management Servlet Protected by Access Gateway When Password Expires” on page 295

URL Parameters

When you are defining the URL for the password service on the Contracts page, the following optional tags can be used in the parameter definitions of the URL. You need to use parameter names that are understood by the service you have selected to use. The Identity Server does not need to understand these parameters, but the password expiration service needs to understand them.

The table below lists a few common ones. Your service might or might not use these, and might require others.

Parameter	Description
<USERID>	Provides the DN of the user with a password that is expired or expiring.
<STOREID>	Provides the name of the user store that authenticated the user before redirecting the user to the password expiration service.
<RETURN_URL>	Provides the URL at the Identity Server to which the user can be redirected after the password service completes.
action=expire	Causes the password expiration service to behave as though the user's password policy is set to allow the user to reset the password even though the user's policy might be set to show the user a hint. The user sees the page to create a new password rather than seeing a hint for an existing password.

For example:

```
https://someservice.com/path/password?user=<USERID>&store=<STOREID>
&returl=<RETURN_URL>&action=expire
```

NOTE: If you copy and paste this text, ensure that you remove the white space between <STOREID> and &returl.

The Identity Server fills in these values, which results in the following URL:

```
https://someservice.com/path/password?user=joe.novell&store=userstore1&
returl=https://myidp.com/nidp/idff/sso&action=expire
```

Forcing Authentication after the Password Has Changed

The password service can also include parameters on the return URL sent to the Identity Server. The Identity Server understands the following parameter:

Parameter	Description
forceAuth=TRUE	When the user is returned to the Identity Server, this parameter forces the user to authenticate with the new password. This eliminates the possibility of an old password being used in an Identity Injection policy.

The following example sends this parameter with https://testnidp.novell.com:8443 as the base URL of the Identity Server.

```
<form id="externalForm" action='https://testnidp.novell.com:8443/nidp/idff/
sso?sid=0&id=117&forceAuth=TRUE' method="post">
```

When the user is redirected to the password management service URL because of an expired password, the POST data in that redirect contains the sid=<> and id=<> values as part of the value used for the Identity Server return URL.

Grace Logins

If you specify a password service and do not specify a value for the number of grace logins in eDirectory, the contract redirects to the password management service only when the grace login count has reached 0 and the password has expired.

The Identity Server needs to read the value of the grace login attribute in order to properly redirect to the password management servlet. If restricting grace logins is not important to your security model, enable grace logins and set the maximum to 9999 (the equivalent of infinite in most environments). For more information, see TID 3465171 (http://www.novell.com/support/php/search.do?cmd=displayKC&docType=kc&externalId=3465171&sliceId=2&docTypeID=DT_TID_1_1&dialogID=131458644&stateId=0%200%20131454892).

Federated Accounts

A user's password does not expire and grace logins are not decremented when you have the following setup:

- The Identity Server is configured to act as a service provider
- User identification is configured to allow federation
- Federation is set up with SAML 2.0, Liberty, or WS Federation protocols

The password expiration service is not called because the user is not using a password for authentication. The service can only be called when the user's account is defederated. After the user has defederated the account, the next time the user logs in, a password is required and the service is called.

Redirection to Password Management Servlet Protected by Access Gateway When Password Expires

When an Active Directory user with an expired password logs in to an authentication contract with a Password Expiration servlet configured, the user is redirected to the password management URI. If the Password Management portal is protected by Access Manager, the user is prompted again for authentication and is not permitted to login as the user password has expired.

If you want the user to be redirected to the Password Management Servlet, perform the following steps:

1. In the Administration Console, click **Devices** > **Identity Servers** > **Edit** > **Local** > **Methods**.
2. Select the authentication method, which is used by the contract where Password Management Servlet is configured.
3. Add the following property for the method used by contract with Password Expiration servlet:

   Property Name = **ExpiredCheck**

   Property Value = **true**
4. Go to **Identity Servers** > **Edit** > **Local** > **Contracts**.
5. Click the associated contract and then click **Options** > **New**.
6. Select **AUTHENTICATE WITH EXPIRED PASSWORD** in **Property Type** and **true** in **Property Value**.
7. Click **OK** > **Apply**, and then **Update** the Identity Server.

Using Login Redirect URL Parameters

When you are defining the URL for login redirect URL on the Contracts page, the following optional tags can be used in the parameter definitions of the URL. You need to use parameter names that are understood by the service you have selected to use. The login redirect URL must understand the name-value pairs you have defined and will use the resolved values in the redirected URL.

Parameter	Description
<USERID>	Provides the DN of the user with a password that is expired or expiring.
<STOREID>	Provides the name of the user store that authenticated the user before redirecting the user to the password expiration service.
<RETURN_URL>	Provides the URL at the Identity Server to which the user can be redirected after the password service completes.

For example:

```
https://someservice.com/path/password?user=<USERID>&store=<STOREID>
&returl=<RETURN_URL>
```

NOTE: If you copy and paste this text, ensure that you remove the white space between <STOREID> and &returl.

The Identity Server fills in these values, that results in the following URL:

```
https://someservice.com/path/password?user=joe.novell&store=userstore1&
returl=https://myidp.com/nidp/idff/sso
```

In addition to the above three parameters you can also configure other parameters.

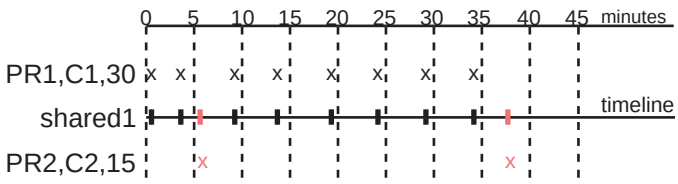
Using Activity Realms

Activity realms are designed to be used with an Access Manager system that uses multiple contracts to protect resources that require different activity timeouts. Activity realms allow you to define how activity at one protected resource affects the activity timeout at another protected resource.

An activity realm essentially represents a time line that tracks the last activity for any resource that is protected by a contract assigned to the activity realm. When a protected resource is accessed, the activity realm associated with the contract is marked as having activity. The contract times out for a protected resource when the elapsed time for activity on the activity realm is greater than the time limit specified in the contract.

For example, suppose you create an activity realm called shared1 and assign it to contract C1 with a timeout of 30 minutes and to contract C2 with a timeout of 15 minutes. Any activity at the resource protected by C1 or C2 marks activity to the shared1 time line. Figure 5-3 illustrates this scenario.

Figure 5-3 *Two Contracts Sharing an Activity Realm*



In Figure 5-3, the user logs into PR1 at time 0, then logs into PR2 at time 6. During the next 30 minutes, the user is active on PR1. The time line for the shared1 activity realm is updated with the user's activity. The user then access PR2 at time 38. Even though no activity has taken place on PR2 for more than the 15-minute contract timeout, PR2 does not time out because activity has occurred within this time at PR1 and because the resources share the same activity realm. Assigning two or more contracts to the same activity realm allows the contracts to influence the timeouts of the other contracts in the activity realm.

When you configure protected resources to use different contracts with different timeouts, they can keep each other alive when they share the same activity realm. If protected resources must not affect each other's activity, they must not share a common activity realm.

You can assign a contract to multiple activity realms. With this configuration, activity on a resource updates the time lines of all activity realms associated with the contract. As long as one of the activity realms has activity within the contract's timeout limit, the user's session remains authenticated.

Activity realms are defined by specifying a name, and the names are case insensitive. Use a comma-separated list to specify multiple names. The system has two default realms that you can use:

- **Any:** Leave the field blank or specify `any` when you want the user's session to remain alive as long as there is some activity by the user at the Access Gateway or at the Identity Server.

  When the Identity Server receives an assertion from another Identity Server that cannot be mapped to a contract, the activity realm is set to `any` with the timeout value equal to the value of the Tomcat session. (The Tomcat session timout is set to the greatest timeout value of the contracts configured for the Identity Server.)
- **NIDPActivity:** Specify `NIDPActivity` for the realm when any activity at the Identity Server by the user can be used to keep the user's session alive.

When you place multiple contracts in the same activity realm, you need to plan carefully so that security limits aren't overruled by activity on less critical protected resources. You also need to carefully balance the desire for single sign-on with the need to require reauthentication for sensitive data. Highly sensitive resources are most secure when they are protected by a contract that is created from its own unique method and that is assigned its own unique activity realm. For more information, see "Assigning a Timeout Per Protected Resource" on page 110.

Specifying Authentication Defaults

You can specify default values for how the system processes user stores and authentication contracts. The default contract is executed when users access the system without a specified contract, and when the Access Gateway is configured to use any authentication.

Additional default contracts can be specified for well-known authentication types that might be required by a service provider. These contracts are executed when a request for a specific authentication type comes from a service provider.

1. In the Administration Console, click **Devices > Identity Servers > Edit > Local > Defaults.**
2. Configure the following fields as necessary:

   User Store: Specifies the default user store for local authentication. If you selected **<Default User Store>** when configuring an authentication method, the system uses the user store you specify here.

   Authentication Contract: Specifies the default authentication contract to be used when users access the Identity Server directly or a protected resource is configured to use **Any Contract**. If you create a new contract and specify it as the default, ensure that you update the Access Gateway configuration if it has protected resources configured to use **Any Contract**.

Authentication Type: Specifies the default authentication contracts to be used for each authentication type. When a service provider requests a specific authentication type, rather than a contract, the identity provider uses the authentication contract specified here for the requested authentication type. For more information, see "Specifying Authentication Types" on page 298.

3 Click **OK**.

4 Update the Identity Server.

Specifying Authentication Types

Trusted service providers can send the Identity Server an authentication request that contains a request for contract or for an authentication type. When the request is for an authentication type, the Identity Server must translate the type to a contract before authenticating the user. You can use the **Authentication Type** section of the Defaults page to specify which contract to use for the common types (classes).

The Identity Server has not implemented all possible types. For types that do not appear on the Defaults page, you can do one of the following:

- You can define a contract for the class whose URI matches the requested class type. When the authentication request is received, the Identity Server uses the URI to match the request with a contract.

  When you create such a contract, you are stating that the contract is security equivalent to the class that is being requested. For configuration information, see "Creating a Contract for a Specific Authentication Type" on page 298.

- You can use the Trust Levels class to assign an authentication level for the requested class. This level is used to rank the requested type. Using the authentication level and the comparison context, the Identity Server can determine whether any contracts meet the requirements of the request. If one or more contracts match the request, the user is presented with the appropriate authentication prompts.

  For configuration information, see "Configuring the Trust Levels Class" on page 149.

Creating a Contract for a Specific Authentication Type

The following steps explain how to create a contract that matches what a trusted service provider is asking for in its authentication request.

1 In the Administration Console, click **Devices > Identity Servers > Edit > Local > Contracts**.

2 To create a new contract, click **New**.

3 Fill in the following fields:

  Display name: Specifies the name of the authentication contract.

  URI: Specifies a value that uniquely identifies the contract from all other contracts. This value must match what the service provider is sending in its authentication request for the type.

  Authentication Level: (Optional) Specify a security level or rank for the contract. This value is not used when authentication request sets the comparison type to exact. It is only used when a contract is selected based on a comparison of authentication levels.

  If the service provider sets the comparison type to minimum, the authentication level can be the same or higher. If the comparison type is set to better, the authentication level must be higher.

  Methods: Select the method that matches the class or type you specified in the URI.

The other fields for the contract are not requirements of the authentication request and can be configured to meet the requirements of the Identity Server. For information about these fields, see “Configuring Authentication Contracts” on page 289.

4 Click **Next**.

5 Configure an authentication card for the contract.

For information about these fields, see “Configuring Authentication Contracts” on page 289.

6 Click **Finish**, then **OK**.

7 Update the Identity Server.

Social Authentication

Access Manager can be configured to support authentication through external OAuth providers like Facebook, Google+, Twitter, LinkedIn, and so on. Social authentication simplifies login for end users and does not require maintaining large user stores. This authentication can be configured using the SocialAuthClass. Login using social identities provide a convenient way for users, improving customer satisfaction and increased registration levels. For more information on how to configure the supported social authentication providers for API Keys and API Secrets, see Appendix , “Configuring Supported Social Authentication Providers for API Keys and API Secrets,” on page 303.

Social login allows business, universities and government entities to leverage social identity providers to share select identity information for authentication via OAuth tokens. This information can then be used to provide protected online services ranging from customer-focused applications, university sites to state and local services and more.

Use Case

Authentication through external OAuth providers can be useful in the following two scenarios:

- Allow external users to access secure resource

  For example, you may want your customers and partners to access https://forums.novell.com. Creating and managing these external users is a hassle for you and the user. Social Authentication helps in this scenario.

  Users will be allowed to sign in with their Facebook or Yahoo ID. Social authentication provider will give Access Manager a set of logged-in user’s attributes. Hence, you will get user data without maintaining it. Access Manager can use this user data and perform actions based on that if required.

- Apply policies to restrict users to access a protected resource

  If the **Identify User Locally** option is selected, the social provider user will be mapped to the local user and you can execute authorization policies based on the user attributes. For example, if Joe is a Facebook user, you can match the attributes of Joe in the local user store based on a rule and execute an authorization policy to access a protected resource. You want to apply policies on an incoming user. For example, your enterprise user 'Bob' has logged into `https://forums.novell.com/`with a social identity. You may want to identify that 'Bob' is your local user and provide him with forum moderator privileges. The **Identify User Locally** option lets you map a social user to your local user and apply appropriate policies.

- **Simplify user login**: You may want to keep the user in your user stores but still make the registration process easy for the users. Social authentication saves the user from remembering another identity. User can login with their social identity while the **Auto Provision User** option will

map the incoming user specified attribute with an existing user in the local user store. If the attribute matches, user will be provisioned, else user will be prompted for local user authentication.

- **Personalized web content in B2C scenarios**: Organizations want to make services and information available in a manner that is personalized to individual. The common approach of creating individual identities for users is costly for the organization and inconvenient for the user. Social login allow users to login with their preferred form of identities. This simplifies the login experience for customers while increasing the registration levels and lowering IT costs.
- **Step up authentication**: While you as an administrator want to improve the user registration through social identities, you would also want to ensure that a second factor authentication is employed when users access sensitive information. Access Manager provides options to configure multiple contracts for protected resources and as users access these resources, they can be prompted to login with a second factor such as their corporate identity or an OTP.

Prerequisite

You must have registered Access Manager with the social authentication providers and must have the API keys and API secrets for establishing federation between Access Manager and the provider for example, Facebook.

Configuring SocialAuthClass

Use the Administration Console to define a new Social Authenticator class, method and contract for the Identity Server cluster. Social authenticator providers such as Facebook, Google+, LinkedIn, and Twitter are supported.

1. Login to the Administration Console.
2. Click **Devices > Identity Servers > Edit > Local > Classes**. Select **New** to add a new class.
3. Specify a name to identify the class. For example, Social authenticator.
4. Select **SocialAuthClass** from the **Java Class** drop-down list. Click **Next**.
5. Configure the **User Identification** settings if you need to perform actions on the logged in user. This is optional. By default, user authentication is done without mapping the social provider user to a local user.
   - **Identify User Locally**: Select this option to map the incoming user to an existing user in your user store. You can apply an authorization policy for these incoming users to provide access control. The following two parameters specify how to perform the user mapping:
     - **Local User LDAP Attribute**: Select an attribute from the drop-down list, for example **LDAP Attribute:mail [LDAP Attribute Profile]**. The incoming configured attribute from the social website is mapped to local user's LDAP attribute.

       NOTE: If there are more than one social authentication providers configured, the **Local User LDAP** attribute must be a multi-valued attribute. This is required to store the social attributes corresponding to each social provider.

     - **Social User Attribute**: Select an attribute which provides a unique user identity for example **Email**. The user email provided in a social website will be mapped to the specified local user's LDAP attribute.

       User mapping is done if the value of **Local User Attribute** is equal to the value of **Social User Attribute**.

NOTE: Provisioning will not occur in the following scenarios:

- If you are going to use Facebook or Google+ as your authentication provider, do not select **DisplayName** as **Social User Attribute** as these providers do not have the **DisplayName** attribute.
- If Social User Attribute is email attribute in Twitter.

- **Auto Provision User**: If you enable this option, incoming user specified attribute will be mapped with an existing user in the local user store. If the attribute matches, user will be provisioned, else user will be prompted for local user authentication. After authentication, user attribute will be mapped and stored.

6 Click **Add** under **Social Auth Providers** to provide the authentication provider details.

- **Auth Provider**: Select the authentication provider from the drop-down list for example, Facebook. You can select from one of the predefined providers or select **Other** to specify your own providers. Note that only the predefined providers have been verified for compatibility with Access Manger. If you select **Other**, you must provide two additional information:
  - **Provider Name**: Specify the name of the provider. Other provider names can be specified under **Others** option. Other provider name can be Yahoo, Hotmail, SalesForce, AOL, FourSquare, MySpace, Instagram, Mendeley or Yammer. Name of social authentication provider is case-sensitive and must match as listed. Else, social authentication class will not work.
  - (Optional) **Implementation Class**: Specify a back end class that can authenticate with these providers if the other providers are not supported. This is needed only for a custom provider that is not in the list provided above.
- **Consumer Key**: Specify the API key that you received when you registered Access Manager with the Social authentication provider.
- **Consumer Secret**: Specify the secret that you received when you registered Access Manager with the Social authentication provider.

7 Click **OK** and **Finish**.

8 Continue with creating a contract and method for this class.

NOTE: With the latest Facebook API, the user's email address is no longer shared by default. For social authentication with Facebook in Access Manager, configure the following properties in the Social Auth method:

```
graph.facebook.com.custom_permissions = email
```

For configuration information, see "Configuring Authentication Methods" on page 287 and "Configuring Authentication Contracts" on page 289.

How Social Authentication Works With Access Manager

For completing social authentication, the Identity Server maps the social attribute value in token to the local user attribute value. The local attribute must be set in the following format for the matching to succeed.

```
<socialprovidername>:<social attribute value>
```

For example, consider that the social authentication class properties are set as follows:

- **Identify User Locally**: Enabled
- **Local User LDAP attribute**: Ldap Attribute:mail

- **Social User Attribute**: Email
- **Auto Provision User**: Enabled
- **Social Auth Provider**: Facebook

As the **Auto Provision User** setting is enabled, after authentication in Facebook, user is asked for a one-time local login. During this process, this user's mail attribute is updated with the social attribute value as `facebook:<social-email-address>`. Subsequent logins from the same user will be seamless and user will be identified automatically.

If **Auto Provision User** setting is disabled, for the authentication to succeed, Access Manager will check if local user LDAP attribute mail value is `facebook:<social-email-address>`.

NOTE: The attribute value is set with the provider's name.

Adding Images for Social Authentication Providers

You can add images for social authentication providers such as Facebook, LinkedIn, Twitter, Google+ and so on. For more information about adding images, see "Adding Authentication Card Images" on page 66.

1. In the Administration Console, click **Devices > Identity Servers > Shared Settings > Authentication Card Images**.
2. Click **New**.
3. Fill in the following fields.

   Name: Specify a name for the image.

   Description: Describe the image and its purpose.

   File: Click **Browse**, locate the image file, then click **Open**.

   Locale: From the drop-down menu, select the language for the card or select **All Locales** if the card can be used with all languages.
4. Click **OK**.
5. If you did not specify **All Locales** for the **Locale,** continue with "Creating an Image Set" on page 67.
6. Add all the required images and click **Close**.

   After configuring the Identity Server with the required social authentication provider images, the login page will display those images on the login page. You can select an image and access the social providers you have added when you access the Identity Server URL.

Changing the Social Authentication Icons

The following procedure allows you to change the default icons of social authentication providers.

1. Go to `socialauth_icons.jsp` file located at `/opt/novell/nids/lib/webapp/jsp/`. You can see all the supported providers and their corresponding public URL locations.
2. To change the icon of a particular provider, go to the icon variable name of that provider and replace the existing URL location with required URL location.

   You can similarly change for other icons defined in the jsp file.
3. Restart the Identity Server after changing the jsp file.

Configuring Supported Social Authentication Providers for API Keys and API Secrets

Access Manager requires API Keys and API Secrets from the supported social authentication providers to integrate with these providers. Follow the steps to configure the supported applications and to get keys from the social authentication providers. You can integrate with Facebook, LinkedIn, Twitter, and Google+. For other providers, see “Configuring SocialAuthClass” on page 300.

NOTE: The procedures documented below may not match the Social Networking Providers’ interface when you create an application. If there are any changes, follow the wizard accordingly. The procedure below is for reference purpose and can vary based on provider configuration page.

Integrating Access Manager with Facebook

The following procedure enables you to generate API Key and API Secret with Facebook.

1 Create a Facebook application for community.

- **1a** Log in to Facebook to access the Application (https://developers.facebook.com/apps) page.
- **1b** From the top right corner, click **Create New App**.
- **1c** Fill in the following fields in the Create a new app screen:
  - **Display Name:** Specify a name for web application.
  - **Category:** Select a category from the drop-down list.
- **1d** Click **Create App**.
- **1e** In the Security Check page key in the displayed Captcha text in the **Text in the box** field and click **Submit**.
- **1f** The Dashboard page displays App Name, App ID and App Secret (hidden). Click **Show** to display the **App Secret**.
- **1g** Copy the values of **App ID** and **App Secret** parameters. You will need these values when you configure Facebook with Access Manager.
- **1h** Click **Settings** on the left. In the **Basic** tab, specify **Contact Email** address.
- **1i** Click **Advanced** tab and specify the following details:
  - Deauthorize Callback URL - For example: https://<IDP URL>:<Port Number>/nidp/app
  - Valid OAuth redirect URIs - For example: https://<IDP URL>:<Port Number>/nidp/jsp/socialauth_return.jsp
- **1j** Click **Status & Review**.
- **1k** Select **YES** to make this application and all its live features available. By default, it is selected as **No**.
- **1l** Your application status will change to Green and will be available online.

2 Configure Facebook application Configuration Setting in Access Manager. The **App ID** and **App Secret** will be used by Access Manager to configure Facebook.

Integrating Access Manager with LinkedIn

The following procedure enables you to generate API Key and API Secret with LinkedIn.

1 Create a LinkedIn application for community.

1a Log in to LinkedIn to access the Developer Network (https://www.linkedin.com/secure/developer) page.

1b Click **Create Application**.

1c Fill in the following fields:

- **Company Name:** Specify the name of the company.
- **Name:** Specify the name of the application.
- **Description:** Specify a description of the application.
- **Application Logo:** Upload the logo of the application.
- **Application Use:** Select a use of the application.
- **Website URL:** Specify the URL of the application.
- **Business Email:** Specify your business email address.
- **Business Phone:** Specify your business phone number.
- Accept **Terms of Service** and **Create Application**.

1d Copy the value of **Client ID** and **Client Secret** parameters. These values will be required when you configure LinkedIn providers with Access Manager.

2 Configure LinkedIn application Configuration Setting in Access Manager.

Integrating Access Manager with Twitter

The following procedure enables you to generate API Key and API Secret with Twitter.

1 Create a Twitter application for community.

1a Log in to Twitter to access the Developers (https://dev.twitter.com/) page using Twitter credentials.

1b From the drop-down box on the right, click on **My Applications**.

1c From Twitter Apps page, click **Create New App** to display Create an application page.

1d Fill in the following fields in the **Application details** page:

- Specify the application **Name**, **Description**, **Website** URL, for example, http://Yourdomain:port and **Callback URL**, for example, http://Yourdomain:port/nidp/jsp/socialauth_return.jsp.

1e Select **Yes, I agree** and click **Create your Twitter application**.

1f From the Details page, verify the details of this application.

1g Navigate to **Settings** tab, select **Allow this application to be used to Sign in with Twitter** and click **Update settings**.

1h Navigate to **API Keys** tab, copy the values of **API key** and **API secret** parameter. These values will be required when you configure Twitter providers with the Access Manager.

1i In your access token, click **Create my access token**. A message that your access token has been successfully created appears.

2 Configure Twitter application Configuration Setting in Access Manager. Access Manager uses the **API key** and **API secret** to configure Twitter.

Integrating Access Manager with Google+

The following procedure enables you to generate API Key and API Secret with Twitter.

1 Create a Google+ application for community.

   1a Log in to Google+ API console. Sign in to Google Cloud Platform (https://code.google.com/apis/console) using Google+ credentials.

   1b From **API Project** on the left select **Create** from the drop-down list.

   1c From Create project page, click **Create project**.

   1d On the All Services page, in the left pane, click **API Access**.

   1e To create an application, on the API Access page, click **Create an OAuth 2.0 client ID…**.

   1f In the Create Client ID page, in the **Product name** field, specify a valid name and click **Next**.

   1g On Client ID Settings page, for **Application type**, choose **Web application**, if it is not selected by default.

   1h In **Your site or hostname**, specify the URL of your production server. For example, example.namdemo.com:8443.

   1i Click **Create client ID**. To generate **Client ID** and **Client secret** for your application, click **Edit settings**.

   1j In the Edit client settings page, specify the authorized **Redirect URIs**, enter https://example.namdemo.com:8443/nidp/jsp/socialauth_return.jsp.

   1k In the authorized **JavaScript origins**, enter: https://example.namdemo.com:8443 and click **Update**.

2 Configure Google+ application Configuration Setting in Access Manager. Access Manager uses the **Client ID** and **Client secret** to configure Google+.

Two-Factor Authentication Using Time-Based One-Time Password (TOTP)

This section explains how to use Time-Based One-Time Password (TOTP) as a second authentication factor with Access Manager. TOTP uses a six-digit number (OTP) in addition to first authentication (for example: username, password) to log in to protected services.

The first step is to register the TOTP client with the secret key. This secret key is used for all future login to the website.

Typically, users download and install the TOTP app on their devices. To log in to a website or service that uses two-factor authentication, in addition to the user name and password, the users enter an OTP generated by the TOTP app. Access Manager validates the OTP and authenticates the user.

- “Why Two-Factor Authentication?” on page 306
- “Prerequisite” on page 306
- “Configuring TOTP Class, Method, and Contract” on page 306
- “Registering with TOTP” on page 307
- “Verifying TOTP Configuration” on page 307

Why Two-Factor Authentication?

Two-factor authentication such as TOTP provides additional security for the systems. It works on the principle of granting access based on a knowledge factor (something the user has) and a possession factor (something the user knows). This helps organizations to implement a multi-factor authentication scheme to satisfy regulatory requirements or increase security.

Prerequisite

- Download and install the TOTP app on your device. This app generates an OTP that is later used for authentication.
- TOTP relies on the device time to generate an OTP. So, it is important that the time on your device is accurate.

Configuring TOTP Class, Method, and Contract

1. Log in to the Administration Console.
2. Click **Devices > Identity Servers > Edit > Local > Classes > New** to add a new class.
3. Specify a name to identify the class.
4. Select `TOTPClass` from the **Java Class** option. The **Java class path** is displayed as `com.novell.nidp.authentication.local.TOTPAuthenticationClass`.
5. Click **Apply**. By default, the TOTP class stores the secret key in the Shared Secret store and no further configuration is required.
6. [Optional] You can also store the secret key in an LDAP attribute, file, or memory. Perform the following steps:

   NOTE: File and Memory class implementation are not recommended for production deployment and are only suitable for a single node Identity Server test environment.

   LDAP user attribute: This option stores the secret key on an LDAP attribute of the user object in the user store.

   1. Select the **Properties** tab of the TOTP class configuration. Add a new property to indicate that the secret key must be stored in an LDAP attribute of the user object in the user store.

      Specify the **Property Name** as `SECRET_STORE_CLASS` and **Property Value** as `USERSTORE`.
   2. Add another property to indicate the attribute in which the secret key must be stored.

      Specify the **Property Name** as `SECRET_LDAP_ATTRIBUTE_NAME` and specify the name of any single-valued attribute. For example, you can specify the **Property Value** as `mobile`, `costcentre` etc.

      The secret key is encrypted and stored in the LDAP attribute. If you do not specify any **Property Value**, the secret key is stored in the `carLicense` LDAP attribute.

      NOTE: Do not use a multi-valued LDAP attribute like `email address` as the **Property Value** as the user registration will fail. It is also important to ensure that the LDAP attribute you have specified as the **Property Value** is a non-operational attribute. For example, it is not recommended to use LDAP Attributes like `groupmembership`.

File class: The File class writes the secret key to a file on the Identity Server file system.

1. Select the **Properties** tab of the TOTP class configuration and add a new property to have the user's secret key stored in a file on the file system.
2. Specify the **Property Name** as `SECRET_STORE_CLASS` and **Property Value** as `FILE`.

Memory class: The Memory based class writes the secret key into memory. This memory is transient in nature and therefore the secret key value is lost each time the Identity Server is restarted.

1. Select the **Properties** tab of the TOTP class configuration and add a new property to define the memory-based property where each user's secret key is stored.
2. Specify the **Property Name** as `SECRET_STORE_CLASS` and **Property Value** as `MEMORY`.

7 Click **Devices > Identity Servers > Edit > Local > Methods**. Select **New** to add a new method.

8 Specify a name to identify the method. Select the TOTP class from the list. This links the TOTP class to the authentication method.

9 Deselect the **Identifies User option**. Click **Apply** to save the changes.

10 Select the user store from the **list of Available user stores** and move it to **User store**.

11 You can either use an existing authentication contract or create a new authentication contract. For example, you can add the default `Name/Password – Form` method as the first method and TOTP method as the second method. Click **Apply** to save the changes.

NOTE: If you use TOTP as a post-authentication method in a federation setup, a `JSP file not found` message is displayed and federation fails.

Registering with TOTP

1 Go to NetIQ Identity Server page `http(s)://<idp server >:<port>/nidp`

2 Select the contract where TOTP is configured as the second method for two-factor authentication.

3 Login with the first method.

4 Click the link beside **Please register for two factor authentication** to generate a OTP. Make a note of the secret key displayed.

If you have installed the TOTP client on your device, scan the code. You can also manually enter the secret key in the TOTP mobile client.

After the registration is complete on the TOTP client on your mobile, the OTP is displayed.

Verifying TOTP Configuration

1 Go to NetIQ Identity Server page: `http(s)://<idp server >:<port>/nidp`

2 Select the contract where TOTP is configured as the second method for two-factor authentication.

3 Login with the first method.

After successfully authenticating with the username and password, prompt is displayed to enter the TOTP OTP.

4 Use the TOTP app to generate the OTP and login by using this OTP.

Persistent Authentication

This authentication class stores user session on the browser after successful login. When the user is prompted for authentication subsequently, this class will reuse the saved authentication instead of prompting the user for credentials. The user will be prompted for credentials again only when the cookie lifetime expires. This type of authentication is used only for applications that do not require very high security. To configure persistent authentication, Access Manager provides a class called PersistentAuthClass.

NOTE: You can configure persistent authentication as a standalone method in a contract.

Configuring Persistent Authenticator Class

The following procedure allows you to configure the PersistentAuthClass. Persistent Authentication supports method properties described in the following section.

1. Login to the Administration Console.
2. Click **Devices > Identity Servers > Edit > Local > Classes**.
3. Click **New**, then specify a **Display name** for example, PersistentAuth.
4. Select **PersistentAuthClass** from the **Java Class** drop-down list.
5. Click **New** to create a new authentication class.
6. In the Add property window, specify the following values. Specifying these values are optional.
   - **Property Name:** Specify the name of the property. For more information on the names you can specify here, see “PersistentAuthClass Properties” on page 308
   - **Property Value:** Specify the property value you would like to define here.
7. Click **OK** and **Finish**.
8. Continue with creating a contract and method for this class.

   For configuration information, see “Configuring Authentication Methods” on page 287 and “Configuring Authentication Contracts” on page 289.

PersistentAuthClass Properties

You can set the following class properties in the configuration file.

- **CryptoKey**: This key is used to encrypt the user's information in the cookie. If this value is long and random, the user information will be secure. The value must be at least ten characters. The certificate private key will be used if you do not set this value. The certificate private key will be used if you do not set this value. If you change or update the certificate, the user is re-authenticated.
- **CookieSuffix**: The Cookie Name is derived using this suffix. PA_ is added as a prefix to the cookie name. By default, cookie name is PA_PERSISTENT_AUTH. For example, if you configure the CookieSuffix as PER_AUTH, the Identity server sends cookie with PA_PER_AUTH name at browser.
- **MaxAgeSeconds**: This property will decide the cookie lifetime. Default value is 86400 seconds (1 day). Maximum value is 4294967295 seconds.

- **ParamName**: The name of the HTTP parameter to enable this feature. The default value of the parameter is EnableCookieAuth. If you want to modify the default value of parameter name for example to TestCookieName, follow the procedure given below.
  1. Login to the Identity Server.
  2. Go to `/opt/novell/nids/lib/webapp/jsp`
  3. Open `login.jsp` file using an editor.
  4. Search for `EnableCookieAuth` parameter name and provide the new name as `TestCookieName` in the input tag.
  5. Ensure that you select the **Remember Me** option.
  6. Restart the Identity Server.

     This value is used by the Identity Server to identify if user has enabled **Remember Me** option on the login page.

Customizing Login Page For Persistent Authentication

To enable **Remember Me** option on a custom login page, use the following custom code in the customized JSP file:

```
<td align=left>
<input type="checkbox" name="EnableCookieAuth" value="true" /> <label> Remember Me
</label>
</td>
</tr>
<tr></tr>
<tr> <td> <span class="instructions">       (Do
not check this box if you are using public computer.)</span> </td></tr>
```

NOTE:

- Ensure that you add the MainJSP and JSP in the method property. For more information, see “Specifying Common Class Properties” on page 285.
- This class doesn't replace the form-based contract. For more information, see “Customizing the Identity Server Login Page” on page 193.

Logging Out of the Persistent Sessions

When a user performs an explicit logout, the Identity Server clears the persistent authentication cookie at browser if the logout request goes through the browser. If SOAP communication is used between the service provider and the Identity server, then the Identity server does not clear the cookie automatically. The cookie can only be cleared by sending a request to a page on the server that issued it. If the page is available on the Identity server, the `clearCookieAuth.jsp` file clears the page. You must customize the service provider’s logout page to run the Identity server’s `clearCookieAuth.jsp` page.

The `clearCookieAuth.jsp` file clears it. The URL for this page will be https://idpserver.example.com/nidp/clearCookieAuth.jsp. Any request to that URL will clear the authentication cookie.

With this class in use, the user will be unable to logout of the system because re-accessing any protected page will simply re-authenticate the user using the user information stored in the browser stored. There are at least two ways to invalidate an outstanding browser stored authentication cookie.

The first is to change the user's password and second is to clear the stored cookie from the browser. Only way to invalidate the cookie is to change the encryption key used. The cookie that is created can only be cleared by a request from the server which created it.

The following configurations are specific to the Novell service provider. If the users uses third party service provider, then the user must customize the logout pages.

In a federation scenario add the following to the `logoutSuccess.jsp` file at `/opt/novell/nam/idp/webapps/nidp/jsp/` of the service provider. You can have logout page redirect to this page, or have an <iframe> that references the page. You may also customize the `/opt/novell/nam/mag/webapps/nesp/jsplogoutSuccess.jsp` file to provide login links or instructions to your user.

```
<tr>
   <td> <iframe src="https://idp.labs.com:8443/nidp/jsp/clearCookieAuth.jsp"
width="0" height="0"> </td>
</tr>
```

where idp.labs.com is the URL of the Identity Server.

Limitations

Following are the limitations with the Persistent Authentication Class:

- User is authenticated even if the password is changed.
- If the user is already logged in with **Remember Me** option enabled, you will be unable to stop the session until the cookie expires.

RADIUS Authentication

RADIUS enables communication between remote access servers and a central server. Secure token authentication through RADIUS is possible because Access Manager works with Novell Modular Authentication Service (NMAS) RADIUS software that can run on an existing NetWare server. Access Manager supports both PIN and challenge-and-response methods of token-based authentication. In other words, RADIUS represents token-based authentication methods used to authenticate a user, based on something the user possesses (for example, a token card). Token challenge-response is supported for two-step processes that are necessary to authenticate a user.

1. In the Administration Console, click **Devices > Identity Server > Edit > Local > Classes**.
2. Click **New**.
3. Specify a display name, then select **RadiusClass** or **ProtectedRadiusClass** from the drop-down menu.
4. Click **Next**.
5. Click **New** to add an IP address for the RADIUS server. You can add additional servers for failover purposes.
6. Click **OK**.
7. Fill in the following fields:

   Port: The port of the RADIUS server.

   Shared Secret: The RADIUS shared secret.

   Reply Time: The total time to wait for a reply in milliseconds

   Resend Time: The time to wait in milliseconds between requests.

   Server Failure Retry: The time in milliseconds that must elapse before a failed server is retried.

JSP: Specify the name of the login page if you want to use something other than the default page. The filename must be specified without the JSP extension. The default page is used if nothing is specified.

User Look Attribute Name: Specify the LDAP attribute on which the user will be searched in the Radius server. CN is the default attribute.

Require Password: Select to require the user to also specify an LDAP password.

8 Click **Finish**.

9 Create a method for this class.

For instructions, see "Configuring Authentication Methods" on page 287.

10 Create a contract for the method:

For instructions, see "Configuring Authentication Contracts" on page 289.

If you want the user's credentials available for Identity Injection policies and you did not enable the **Require Password** option, add the password fetch method as a second method to the contract. For more information about this class and method, see "Password Retrieval" on page 325.

11 Update the Identity Server.

Client Integrity Check

You can configure a client integrity check policy to verify the recommended software (such as firewall and antivirus software) are installed on the client machine. You can configure different policies for Windows, Linux, and Macintosh machines and specify software that must be available in client machines to pass the client integrity check.

You need to create an identity provider authentication class that checks for the specified software on the client machine. You can configure policies to check processes, files, Windows registry, system services, and so on. This class can be executed with the first method of the contract. If the check fails, the user authentication fails.

Configuring Client Integrity Check

1 Download the Client Integrity Check (CIC) package from the CIC Package download page and extract the `CIC_utility.tar.gz` file.

Specifying details for the Operating System

2 Traverse to `CICtool/conf/config.xml` file. Add the following details to the `config.xml` file:

```
<OperatingSystem Name="Linux" UserInterfaceID="Linux" CICOSID="Linux"> . . . </
OperatingSystem>
```

To define operating system details for Windows and Macintosh, substitute `UserInterfaceID`, `Name` and `CICOSID` with *Macintosh* or *Windows*.

NOTE: The attribute `Name` indicates the identifier for the operating system. Ensure that you use the same identifier for `UserInterfaceID` and `CICOSID`.

Adding a Category

A category is a group of similar software. For example, a firewall category can contain a list of firewalls such as the Windows Firewall and ZoneAlarm firewall. You can configure multiple software categories under each operating system in single cic policy.

When multiple categories are configured for an operating system, if one of the enabled category does not exist on the client, the client integrity check fails.

3 A category can be added to a operating system by adding <Type> tag in config.xml as follows:

```
<OperatingSystem Name="Linux" UserInterfaceID="Linux" CICOSID="Linux"> <Type
Name="Firewall_Linux" UserInterfaceID="Firewall_Linux" Status="true"
CICTypeID="Firewall_Linux"> ... </Type> <Type Name="Antivirus_Linux"
UserInterfaceID="Antivirus_Linux" Status="true" CICTypeID="Antivirus_Linux"> .
</Type></OperatingSystem>
```

As described in this example, multiple categories can be configured under an operating system.The Name attribute inside the <Type> tags indicate the category name.

Set status to *true* to enable a specific category.

Adding Applications for a Category

A category consists of group of applications. You can add more than one application under a category. A client workstation is checked for the presence of any one of the software items in the category. If at least one of the enabled application definition exists on the system, the client integrity check passes for that category.

4 To configure applications to a category, add <Info> tag as shown below.

```
<OperatingSystem Name="Linux" UserInterfaceID="Linux" CICOSID="Linux"> <Type
Name="Firewall_Linux" UserInterfaceID="Firewall_Linux" Status="true"
CICTypeID="Firewall_Linux"> <Info Name="FireStarter"
UserInterfaceID="FireStarter" Status="true"> . . . </Info> </Type> <Type
Name="Antivirus_Linux" UserInterfaceID="Antivirus_Linux" Status="true"
CICTypeID="Antivirus_Linux"> <Info Name="AntiVir" UserInterfaceID="AntiVir"
Status="true"> . . . </Info> </Type></OperatingSystem>
```

The Name attribute inside the <Info> tags indicate the application name. Set status to *true* to enable a specific application.

NOTE: To enable an application you must have already enabled the category that it belongs to.

Adding Attributes for an Application

After you have added an application to a category, you must configure the attributes for each of these applications. These attributes can be in the form of RPMs, processes, registry keys, or executable files. The client integrity check detects the presence of these attributes.

5 These attributes can be configured under each application by adding attribute type tags to config.xml as follows:

```
 <OperatingSystem Name="Linux" UserInterfaceID="Linux" CICOSID="Linux"> <Type
Name="Firewall_Linux" UserInterfaceID="Firewall_Linux" Status="true"
CICTypeID="Firewall_Linux"> <Info Name="FireStarter"
UserInterfaceID="FireStarter" Status="true"> <AbsoluteFile UserInterfaceID="0"
Name="/var/lock/subsys/firestarter" HashMD5="" /> <RPM UserInterfaceID="1"
Name="FireStarter" Version="0.9.3" /> </Info> </Type> <Type
Name="Antivirus_Linux" UserInterfaceID="Antivirus_Linux" Status="true"
CICTypeID="Antivirus_Linux"> <Info Name="AntiVir" UserInterfaceID="AntiVir"
Status="true"> <Process UserInterfaceID="0" Name="antivir" Owner="root" />
<AbsoluteFile UserInterfaceID="1" Name="/usr/lib/AntiVir/avguard" HashMD5="ss"
/> </Info> </Type> </OperatingSystem>
```

In this example, <AbsoluteFile>,<RPM>,<Process>,<AbsoluteFile> are examples of attribute type tags and fields like *Name*,*Version*,*Owner* are examples of attribute names.

For more information about attributes for applications on different operating systems, see "Configuring Attributes for an Application" on page 315

Client Security Levels

6 You can configure different levels of client security. For more information about the different levels of client security, see “Client Security Levels” on page 315

The security level can be configured by adding the following details to the `config.xml` file:

```
<SecurityLevel Name="None" UserInterfaceID="None" DisplayMessage="Client
Integrity failed" SecurityLevelID="None" CICReferenceCount="0"
TrafficReferenceCount="1" /><SecurityLevel Name="Low" UserInterfaceID="Low"
DisplayMessage="Your workstation is at Least Secure Level" SecurityLevelID="1"
CICReferenceCount="3" TrafficReferenceCount="1"> . . .</SecurityLevel>
```

The value of the `Name` field can be *None*,*Low*,*Moderate* and *High*,and the `SecurityLevelID` value in each case must be *None*,*1*,*2* and *3* respectively.

Adding Operating System details to the Security Level

7 Under each security level, an operating system can be configured by adding `<CICOS>` tag as follows:

```
<SecurityLevel Name="Low" UserInterfaceID="Low" DisplayMessage="Your
workstation is at Least Secure Level" SecurityLevelID="1" CICReferenceCount="3"
TrafficReferenceCount="1"> <CICOS UserInterfaceID="Linux" CICOSIDRef="Linux">
. </CICOS> <CICOS UserInterfaceID="Windows" CICOSIDRef="Windows"> . </CICOS>
<CICOS UserInterfaceID="Macintosh" CICOSIDRef="Macintosh"> . . . </CICOS></
SecurityLevel>
```

This example shows configuration of operating system for security level *Low*,Other levels can be incorporated in the same manner.

8 Under each operating system, category can be configured by adding `<CICType>` tag as follows:

```
<SecurityLevel Name="Low" UserInterfaceID="Low" DisplayMessage="Your
workstation is at Least Secure Level" SecurityLevelID="1" CICReferenceCount="3"
TrafficReferenceCount="1"> <CICOS UserInterfaceID="Linux" CICOSIDRef="Linux">
<CICType UserInterfaceID="Firewall_Linux" CICTypeIDRef="Firewall_Linux"
CICTypeStatus="true" /> <CICType UserInterfaceID="Antivirus_Linux"
CICTypeIDRef="Antivirus_Linux" CICTypeStatus="true" /> </CICOS> <CICOS
UserInterfaceID="Windows" CICOSIDRef="Windows"> <CICType
UserInterfaceID="Firewall_Windows" CICTypeIDRef="Firewall_Windows"
CICTypeStatus="true" /> <CICType UserInterfaceID="Antivirus_Windows"
CICTypeIDRef="Antivirus_Windows" CICTypeStatus="true" /> </CICOS> <CICOS
UserInterfaceID="Macintosh" CICOSIDRef="Macintosh"> <CICType
UserInterfaceID="Antivirus_Mac" CICTypeIDRef="Antivirus_Mac"
CICTypeStatus="true" /> </CICOS></SecurityLevel>
```

9 Traverse to the `CIC/CICtool/bin` directory. Execute the `CICtool` binary by using the following command:

`$./CICtool ../conf/config.xml`.

This creates `.txt` policy files in the `CICtext` folder.

10 In the Identity Server, create the following directories by using the following commands:

11 From the `CICtext` directory, copy the `cic_linux.txt`, `cic_mac.txt` and `cic_windows.txt` to the respective `CIC` system directory created in step 10.

Use the following command to copy:

Substitute idp login credentials with the server *IPaddress*, *port, username* and *password* to login to the Identity Server.

12 From the CIC bin directory, copy the `LinCic`, `MacCic` and `wincic.exe` to the respective CIC system directory created in step 10.

Use the following commands to copy:

Substitute idp login credentials with the server IPaddress, port, username and password to login to the Identity Server

13 In the Administration Console, click **Identity Server** > **Edit** > **Local** > **Classes** > **New**

14 Specify a name for the class and select `ClientIntegrityCheckClass` in **Java class**. Click **Next**.

15 Click **New** and specify the following property name and property value:

Name	Value
`windowsBinary`	`/nidp/classUtils/windows/wincic.exe`
`windowsPolicy`	`/nidp/classUtils/windows/cic_windows.txt`
`linuxBinary`	`/nidp/classUtils/linux/LinCic`
`linuxPolicy`	`/nidp/classUtils/linux/cic_linux.txt`
`maci386Binary`	`/nidp/classUtils/mac/MacCic`
`maci386Policy`	`/nidp/classUtils/mac/cic_mac.txt`

16 Click **OK** > **Finish**.

17 Create a method for this class and deselect **Identifies User** check box and set all other fields to default settings and click **OK**. For instructions, see “Configuring Authentication Methods” on page 287.

18 Go to the **Contracts** tab and select CIC method from the **Available Methods** list and click **OK**. For instructions, see “Configuring Authentication Contracts” on page 289.

Client Security Levels

You can configure the level of security configured at the client machine. You can decide the categories of software that you want to be present for each level.

You can configure the following security levels:

- **Least Secure:** Specifies the minimum categories of software that must be present on a client machine for the client to be at the lowest secure level. When a client is at a least secure level, you can configure the traffic policies so that the client has access to limited set of resources.
- **Moderately Secure:** Specifies the categories of software that must be present on a client machine for the client to be at a moderately secure level. When a client is at a moderately secure level, you can configure the traffic policies accordingly.
- **Secure:** Specifies the software categories that must be present on a client machine for the client to be secure. When a client is at a secure, the traffic policies can be configured so that the client has access to all or most of the protected resources, depending on the role of the client.
- **None:** If a client does not have any of the software such as firewall or antivirus specified in the client integrity check policy, then the security level of that client is None. When a client is at this level, the SSL VPN connection is established, but the client is given access to only a minimal set of resources.

Configuring Attributes for an Application

Specify details for the attributes. The following table lists the attributes for applications on different operating systems:

Operating System	Attribute Type	Attribute Name
Linux	RPM	**Name:** Specify the name of the RPM that must be present on the client machine. **Version:** Specify the version of the RPM that must be present on the client machine.
	Process	**Name:** Specify the name of the process that must be present on the client machine. **Owner:** Specify the owner of the process.
	Absolute File	**Name:** Specify the name and absolute path of the file that must be present on the client machine. **HashMD5:** Specify the MD5 checksum value of the absolute file. To calculate the MD5 checksum value of an absolute file located in your local system, click **Select File** to select the file. The MD5 checksum value of the selected file is displayed. To calculate the MD5 checksum value for an absolute file that is on another system, remotely connect to that system, calculate the MD5 value, then copy the value in the **HasMD5** field. **NOTE:** You can also copy the file from the remote system to the local system, then calculate the MD5 checksum by using the **Select File** option. However, this might change the MD5 value of the file during the process. If you want to use this method, then ensure that the file size and file contents did not change during the process.
Macintosh	Package	**Name:** Specify the name of the software package that must be present on the client machine. **Version** Specify the version of the software package.
	Process	**Name:** Specify the name of the executable file that must be present on the client machine. **Owner:** Specify the owner of the process.
	Absolute File	**Name:** Specify the name and absolute path of the file that must be present on the client machine. **HashMD5:** Specify the MD5 checksum value of the absolute file. To calculate the MD5 checksum value of an absolute file located in your local system, click **Select File** to select the file. The MD5 checksum value of the selected file is displayed. To calculate the MD5 checksum value for an absolute file that is on another system, remotely connect to that system, calculate the MD5 value, then copy the value in the **HasMD5** field. **NOTE:** You can also copy the file from the remote system to the local system, then calculate the MD5 checksum by using the **Select File** option. However, this might change the MD5 value of the file during the process. If you want to use this method, then ensure that the file size and file contents did not change during the process.

Operating System	Attribute Type	Attribute Name
Windows	Process	**Name:** Specify the name of the executable file that must be present on the client machine. **RegistryKeyName:** Specify the registry key name. When you add this name, make sure that you also specify a value for **RegistryKey Value**. **ValueName:** Specifies the value for RegistryKey configured. The data found in this key value must be the absolute path of the folder where the process file is present. **Version:** Specify the version of the software process that must be running in the client machine. **NOTE:** The version attribute specifies the Windows Explorer file version number.
	RegistryKey	**Name:** Specify the name and absolute path of the registry key that must be present on the client machine. **Value Name:** Specify the name of the registry key value. **Value Data:** Specify a data for the registry key value. This data can be for registry type REG_BINARY, REG_DWORD, REG_DWORD_LITTLE_ENDIAN, REG_MULTI_SZ, or REG_SZ. The value for REG_DWORD and REG_DWORD_LITTLE_ENDIAN is hexadecimal or decimal. The value of a REG_MULTI_SZ or REG_SZ can be a string value or, numeric or alphanumeric. The value of REG_BINARY can be binary or hexadecimal. The Value name and Value data are separated by a comparison operator such as `=, >. <, <=, >=.` You must always use `=` with a string or with the registry type REG_BINARY. You can use any comparison operator with other registry types For example, if the registry key name is specified as `RegKey` with a Value Name of `RegValue`, a comparison operator of =, and a Value Data of `RegData`, the client integrity check process looks for the presence of `RegKey` with a value name `RegValue` = value data `RegData` on the client machine. If the registry is present with the specified values, the client passes the client integrity check. **NOTE:** Registry keys are not case sensitive, and they can contain either a single backslash (\) or double backslash (\\). For example: One of the registry key descriptions is HKEY_Local_Machine\\Software\\Symantec. It can also be written as HKEY_Local_Machine\Software\Symantec.

Operating System	Attribute Type	Attribute Name
	Absolute File	**Name:** Specify the name and absolute path of the file that must be present on the client machine. **Version:** Specify the version of the absolute file that must be running on the client machine. **HashMD5:** Specify the MD5 checksum value of the absolute file. To calculate the MD5 checksum value of an absolute file located in your local system, click **Select File** to select the file. The MD5 checksum value of the selected file is displayed. To calculate the MD5 checksum value for an absolute file that is on another system, remotely connect to that system, calculate the MD5 value, then copy the value in the **HasMD5** field. **NOTE:** You can also copy the file from the remote system to the local system, then calculate the MD5 checksum by using the **Select File** option. However, this might change the MD5 value of the file during the process. If you want to use this method, then ensure that the file size and file contents did not change during the process.
	Service	**Name:** Specify the display name of the service. **Status:** Specify the status of the process in the client machine. The status of the process can be **Running** or **Stopped**.

Mutual SSL (X.509) Authentication

Mutual authentication is used when a user is issued an X.509 certificate from a trusted source, and the certificate is then used to identify the user. To ensure the validity of the certificates, Access Manager supports both Certificate Revocation Lists (CRLs) and Online Certificate Status Protocol (OCSP) methods of verification.

To configure X.509 authentication, you need to create an authentication class, then configure the validation and attribute mapping options.

1. Log in to the Administration Console.
2. Import the trusted root certificate or certificate chain of the Certificate authority into the Identity Server trusted root store.

   For information about how to import trusted roots, see “Importing Public Key Certificates (Trusted Roots)” on page 815.

   The Identity Server must trust the Certificate authority that created the user certificates.
3. To create the X.509 authentication class, click **Devices > Identity Servers > Edit > Local > Classes**.
4. Click **New**.
5. Specify a display name, then select **X509Class** from the drop-down menu.
6. Click **Next**.
7. Configure the validation options:

Access Manager caches CRLs, so the revoked status of a newly revoked certificate is not picked up until the next cache refresh. For higher security requirements, use OCSP validation with CRL validation. You can select None, CRL, OCSP, OCSP-CRL, or CRL-OCSP validation. In a production environment, for highest security, select either OCSP-CRL or CRL-OCSP validation. The default setting is to check OCSP first, then CRL.

CRL Validation: Checks the CRL. If you enable CRL validations, the CRL distribution point extension is read out of the user's X.509 certificate. The CRL distribution point contains the URL where the complete CRL can be found, as published by the certificate authority. The system checks the CRL itself and then checks to see if the user certificate is in the revoked list. The system can get the CRL over HTTP and LDAP. If you are not expecting the distribution point in user certificates, you can specify a value in the **LDAP URL** option to get the CRL.

Access Manager supports two schemes for a URL: `http://` and `ldap://`.

OCSP Validation: If OCSP validation is enabled, the Authority Info Access point (AIA) is read out of the user certificate, which contains the URL for the OCSP responder. A signed OCSP request for the user certificate is sent to OCSP responder. A signed OCSP response is received from the responder that has the revoked status for the user certificate. Alternately, if you are not expecting an AIA in a user certificate, you can specify a value in the OCSP responder **URL** field. The value you enter here overrides any OCSP responder URLs in a certificate.

Access Manager supports two schemes for a URL: `http://` and `ldap://`.

Disable Root CA Revocation Check: Disables whether to check if a certificate authority has been revoked. This option checks the CRL and OCSP for the trusted root certificate in the chain. You can enable or disable this option for X.509 user authentication performance.

If you enable the root CA revocation check, what the Identity Server checks depends upon the certificates that have been added to the Identity Server trust store. If the root certificate and the intermediate certificates in the chain are in the trust store, the Identity Server only validates the client (leaf) certificate. If the trust store only contains the root certificate, the browser sends the intermediate and leaf certificates, which are then validated by the Identity Server.

8 Configure the browser restart option.

Some browsers, such as Internet Explorer, keep the SSL session active until the user closes the browser. When the user logs in with the certificate on a smart card, then removes the card and logs out but does not close the browser, the SSL session is still active. If another user has access to the machine, that user can use the existing session.

To prevent this from happening, select **Force browser restart on logout**.

9 Click **Next**.

10 Continue with "Configuring Attribute Mappings" on page 319.

Configuring Attribute Mappings

The attribute mapping options allow you to specify how the Identity Server maps the certificate to a user in the user store. **Subject name** is the default map.

1 Step 3 of the wizard or click **Devices > Identity Servers > Edit > Local > Classes** > **[Name of X.509 class]** > **Properties** > **Attributes**.

2 Configure attribute mappings.

Show certificate errors: Displays an error page when a certificate error occurs. This option is disabled by default.

Auto Provision X509: Enables using X.509 authentication for automatic provisioning of users. This option allows you to activate X.509 for increased security, while using a less secure way of authentication, such as username/password. Extra security measures can even include manual intervention to activate X.509 authentication by adding an extra attribute that is checked during authentication.

An example of using this option is when a user authenticates with an X.509 certificate, a lookup is performed for a matching SASallowableSubjectNames with the name of the user certificate. When no match is found, and **Auto Provision X509** is enabled, the user is presented with a custom error page specifying to click a button to provide additional credentials, such as a username and password, or to start an optional Identity Manager workflow. If the authentication is successful, then the user's SASallowableSubjectNames attribute is filled in with the certificate name of the user certificate.

When **Auto Provision X509** is enabled, and the attribute that is used for subject name mapping is changed from the default sasAllowableSubjectNames, you need to ensure that the LDAP attribute that is used can store string values with a length as long as the longest client certificate subject name. For example, if you use the LDAP attribute title (which has an upper bound of 64 characters) the **Auto Provision X509** fails the provisioning part of the authentication if the client certificate subject name is longer 64 characters. The authentication works if a valid name and password is given. However, provisioning fails.

Attributes: The list of attributes currently used for matching. If multiple attributes are specified, the evaluation of these attributes must resolve to only one user in the user store.

The evaluation first does a DN lookup for subject name or directory name mapping. If this fails, the rest of the mappings are looked up in a single LDAP query.

Available attributes: The available X.509 attributes. To use an attribute, select it and move it to the **Attributes** list. When the attribute is moved to the **Attributes** list, you can modify the mapping name in the **Attribute Mappings** section. The mapped name must match an attribute in your LDAP user store.

Directory name: Searches for the directory address in the client certificate and tries to match it to the DN of a user in the user store. If that fails, it searches the sasAllowableSubjectNames attribute of all users for a value that matches. The sasAllowableSubjectNames attribute must contain values that are comma-delimited, with a space after the comma. (For example, O=CURLY, OU=Organization CA or OU=Organization CA, O=CURLY.)

Email: Searches for the email attribute in the client certificate and tries to match it with a value in the LDAP mail attribute.

Serial number and issuer name: Lets you match a user's certificate by using the serial number and issuer name. The issuer name and the serial number must be put into the same LDAP attribute of the user, and the name of this attribute must be listed in the **Attribute Mappings** section.

When using a Case Ignore String attribute, both the issuer name and the serial number must be in the same attribute separated by a dollar sign ($) character. The issuer name must precede the $ character, with the serial number following the $ character. Do not use any spaces preceding or following the $ character. For example: O=CURLY, OU=Organization CA$21C0562C5C4

The issuer name can be from root to leaf or from leaf to root. The issuer name must be comma-delimited with a space after the comma. (For example, O=CURLY, OU=Organization CA or OU=Organization CA, O=CURLY.)

The serial number cannot begin with a zero (0) or with a hexadecimal notation (0x). If the serial number is 0x0BAC05, the value of the serial number in the attribute must be BAC05. The certificate number is displayed in Internet Explorer with a space after every fourth digit. However, you must enter the certificate number without using spaces.

The LDAP attribute can be any Case Ignore List or Case Ignore String attribute of the user. If you are configuring your own attribute, ensure that the attribute is added to the Person class. When using a Case Ignore List attribute, both the issuer name and the serial number must be in the same list. The issuer name needs to be the first item in the list, with the serial number being the second and last item in the list.

Subject name: Searches for the Subject name of the client certificate and tries to match it to the DN of a user in the user store. If that fails, it searches the sasAllowableSubjectNames attribute of all users for a value that matches the Subject name of the client certificate. The sasAllowableSubjectNames attribute must contain values that are comma-delimited, with a space after the comma. (For example, O=CURLY, OU=Organization CA or OU=Organization CA, O=CURLY.)

3 Click **Finish**.

4 Create a method for this class.

For instructions, see “Configuring Authentication Methods” on page 287.

5 Create a contract for the method:

For instructions, see “Configuring Authentication Contracts” on page 289.

If you want the user’s credentials available for Identity Injection policies, add the password fetch method as a second method to the contract. For more information about this class and method, see “Password Retrieval” on page 325.

6 Update the Identity Server.

Setting Up Mutual SSL Authentication

SSL provides the following security services from the client to the server:

- Authentication and nonrepudiation of the server, using digital signatures
- Data confidentiality through the use of encryption
- Data integrity through the use of authentication codes

Mutual SSL provides the same things from the server to the client as SSL. It provides authentication and nonrepudiation of the client, using digital signatures.

1 Set up Access Manager certificates for security, and import them into the Access Manager system. (See Section 11, “Creating Certificates,” on page 799.)

2 Create an X.509 authentication class. (See “Mutual SSL (X.509) Authentication” on page 318.)

3 Create an authentication method using this class. (See “Configuring Authentication Methods” on page 287.)

4 Create an authentication contract using the X.509 method. (See “Configuring Authentication Contracts” on page 289.)

5 Update the Identity Server cluster configuration. (See “Updating an Identity Server Configuration” on page 191.)

6 Update any associated Access Gateways to read the new authentication contract.

7 Assign the contract to protect resources.

See “Configuring Protected Resources” on page 101.

8 Update the Access Gateway.

Customizing Certificate Errors

In case of certificate validation failure, the browser displays a standard `Page expired` error. If you want the Identity Server to display an Access Manager error instead of the usual error messages provided by the browser, edit the `/opt/novell/nam/idp/conf/server.xml` by using the following procedure:

1. Search for the `clientauth` attribute in the `server.xml` file.
2. Modify the value of the `clientauth` attribute from the default value of `false` to `want`.

   NOTE: if you use `cliantauth=want` in a connector, ensure that the connector also contains `protocol="org.apache.coyote.http11.Http11Protocol"`.

3. Save the file and restart Identity Server by using the `rcnovell-idp restart` command.

   This setting ensures that the certificate is exchanged between the client and the server. This will result in a prompt being displayed on the browser during authentication.

4. Export the user and server certificate from the Administration Console by using the **Security > Certificates** option.

   To avoid the untrusted certificate messages in browsers, import the trusted root certificate of the CA into your browsers. For details, see "Resolving Certificate Import Issues" on page 1000.

Configuring X.509 Authentication to Provide Access Manager Error Message

You can configure the X.509 class property and Identity Server to avoid the browser provided message and display Access Manager error message. This occurs when the SSL mutual handshake fails because of non availability of client certificate.

NOTE: You can specify the Port and URL name as per your environment. The URL name and port number specified in the following procedure is an example.

Prerequisite: To configure you must have a parent domain, for example, https://240onbox.provo.novell.com:8443/nidp/ and sub domain https://onbox.provo.novell.com:8448/ available. You can also have a different port or IP combination.

Follow the procedure given below to configure X.509 based authentication for Access Manager specific error.

1. Go to identity provider and navigate to `/opt/novell/nam/idp/conf` directory.
2. Open `server.xml` file using the vi editor.
3. Search for `<Connector NIDP_Name="connector"` and create a copy of existing connector.
4. Go to the new connector you created and search for `clientAuth=false` string andchange it to `clientAuth=want`. Change the port to a new port for example, 8448.
5. Save the `server.xml` file and exit.
6. Navigate to `/opt/novell/nids/lib/webapp/META-INF/` directory and open `context.xml` file.
7. Change Tomcat `context.xml` to set a same cookie for sub domains. Ensure that the path is set to `"/"` as follows.

   ```
   <?xml version="1.0" encoding="UTF-8"?> <Context sessionCookiePath="/"
   sessionCookieDomain=".provo.novell.com"> <!-- Disable session persistence
   across Tomcat restarts --> <Manager pathname="" saveOnRestart="false"/> </
   Context>
   ```

8 Change session proxying for setting this cookie.

8a Navigate to **Devices** > **Identity Servers** > **Edit** > **Options**.

8b Click **New**. Specify the following details:

Property Name: Select CLUSTER COOKIE DOMAIN

Property Value: Specify.provo.novell.com

Property Name: Select CLUSTER COOKIE PATH

Property Value: Specify /

NOTE: Before you go to the next step, ensure that you have configured X.509 class, method, and contract. For more information, see "Mutual SSL (X.509) Authentication" on page 318.

---

9 For X509Class based redirection, in the X.509 method add the property for X.509 to be redirected to the new connector as follows.

```
Property Name: CONNECTOR_HOST Property Value: https://
onbox.provo.novell.com:8448/
```

Here, the DNS name `onbox` is a sub domain as indicated in the prerequisite.

10 Restart Tomcat using the following commads.

- Linux: `/etc/init.d/novell-idp restart`
- Windows: Enter the following commands:
  - `net stop Tomcat7`
  - `net start Tomcat7`

Verify the configuration as follows:

Access the Identity Server URL in browser that does not have client certificate. Access the X.509 authentication card and verify the behavior. It must redirect to connector port 8448 and redirect back to original page with an Access Manager error message or error code.

ORed Credential Class

Access Manager includes a class that can be configured to accept any combination of name/password, X.509, or RADIUS credentials. When this class executes as part of a contract, users can select and enter their preferred type of credential.

For example, if a name/password credential is ORed with an X.509 credential, the user can select to use a certificate or to enter a name and password. As an administrator, you have decided that both credentials are equally secure for the protected resource the contract is protecting.

To create an ORed credential class:

1 In the Administration Console, click **Devices** > **Identity Servers** > **Edit** > **Local** > **Classes**.

2 Click **New**, then fill in the following fields:

Display name: Specify a name for the class.

Java class: Select NPOrRadiusOrX509Class.

3 Click **Next**, then select the types of classes you want to OR. You must select at least one of the following:

Use Name/Password: Select this option if you want the PasswordClass to be one of the authentication options available to the user.

Use Radius: Select this option if you want the RadiusClass to be one of the authentication options available to the user.

Use X509: Select this option if you want the X509Class to be one of the authentication options available to the user.

4 (Conditional) If you want to use the protected version of the PasswordClass or RadiusClass, select the **Enforce use of HTTPS** option.

5 (Conditional) If you selected the **Use Name/Password** option, configure the properties:

5a In the **Name/Password Properties** section, click **New**.

5b Specify a property name and property value.

For information about the properties that the PasswordClass and the ProtectedPasswordClass support, see "Specifying Common Class Properties" on page 285.

5c Click **OK**.

5d Repeat Step 5a through Step 5c to add more than one property.

6 Click **Next**.

7 (Conditional) If you selected the **Use Radius** option, configure the Radius properties.

For information about the configuration options, see "RADIUS Authentication" on page 310.

8 (Conditional) If you selected the **Use X509** option, configure how the certificate is validated.

For information about the configuration options, see "Mutual SSL (X.509) Authentication" on page 318.

9 Click **Next**.

10 (Conditional) If you selected the **Use X509** option, configure the attribute mappings.

For information about the configuration options, see "Mutual SSL (X.509) Authentication" on page 318.

11 Click **Next**.

12 Click **Finish**.

13 Continue with creating a method and a contract for this class.

For configuration information, see "Configuring Authentication Methods" on page 287 and "Configuring Authentication Contracts" on page 289.

The Radius class prompts the user for a token instead of a password. The user can use the drop-down menu to select between the password and the token. If the user selects to send a certificate, the username and password/token options become unavailable.

OpenID Authentication

OpenID is an open, decentralized method for identifying users which allows users to use the same digital identity for logging in to multiple services. You can configure the Identity Server to trust the provider or providers of OpenIDs by configuring the OpenID class. You then configure a method and contract and assign a protected resources to use the contract for authentication. When the users supply the OpenID, they are granted access if the Identity Server has been configured to trust the provider of the OpenID server.

NOTE: Access Manager supports OpenID1.1.

---

1 In the Administration Console, click **Devices** > **Identity Servers** > **Edit** > **Local** > **Classes**.

2 Click **New**, then fill in the following fields:

Display name: Specify a name for the class.

Java class: Select OpenIdClass.

The Java class path is configured automatically.

3 Click **Next**, then configure the following properties:

Open ID Provider Substrings: Specify at least one URL substring of an OpenID provider. The OpenID URL that user enters during the login process must contain one of the strings as a subset of the OpenID URL. For example, if user enters `https://user123.myopenid.com`, this field needs to contain one of the following strings:

```
myopenid.com
.myopenid.com
```

To specify multiple URLs, separate them with a semicolon (;)

Identity the OpenID user locally: After the user authenticates at the OpenID provider, Access Manager can associate a username from the user store with the OpenID user. With this association, Access Manager can use the policies defined for the username to enforce access to protected resources.

- When this option is not selected, the OpenID user is not mapped to a local user. The username of the authenticated user remains as the OpenID URL. For example, if the user enters http://user123.myopenid.com for the URL, `http://user123.myopenid.com` becomes the username.
- When this option is selected, an attempt is made to map the OpenID user with a username in the user store. You can do this manually by storing the user's OpenID in the attribute specified in the **LDAP Attribute Name** option. You can also have the Identity Server add the OpenID value to the attribute by selecting the **Auto Provision LDAP Attribute** option.

LDAP Attribute Name: Specify the name of the attribute that contains the identification information for the users. For OpenID authentication, this attribute must contain the OpenID for the user.

Auto Provision LDAP Attribute: Select this option when you want the user to provide additional information for identification for the first authentication, such as a username and password. The Identity Server uses this information to identify the user, then writes the user's OpenID value to the attribute specified in the **LDAP Attribute Name** option. On subsequent logins, the Identity Server can identify the user by using the specified attribute and the user is not prompted for additional information.

4 Click **Finish**.

5 Create a method for this class.

For instructions, see "Configuring Authentication Methods" on page 287.

6 Create a contract for the method:

For instructions, see "Configuring Authentication Contracts" on page 289.

If you want the user's credentials available for Identity Injection policies, add the password fetch method as a second method to the contract. For more information about this class and method, see "Password Retrieval" on page 325.

7 Update the Identity Server.

Password Retrieval

If you have configured contracts that do not use a username and password for the credentials and you want to configure single sign-on to protected resources that require a user's name and password, you need to configure the PasswordFetchClass to retrieve the user's name and password. You need

to create the class, then create a method from the class. The method needs to be assigned as the second method for the authentication contract that does not prompt the user for a username and password. When the Identity Server executes the contract, the PasswordFetchClass retrieves the username and password and stores them with the LDAP credentials, which makes them available for Identity Injection and Form Fill policies.

For example, your contract may be using Kerberos or X.509 certificate authentication where the password is not available. Use the PasswordFetchClass to retrieve the username and password.

IMPORTANT: The PasswordFetchClass only works with eDirectory user stores.

1. Click **Devices** > **Identity Servers** > **Edit** > **Local** > **Classes**.
2. Click **New**, then fill in the following fields:

   Display name: Specify a name for the class.

   Java class: Select **PasswordFetchClass**.

   The Java class path is configured automatically.
3. Click **Next**, then configure the following general properties:

   Ignore password retrieval failure: Select this option if you want users to continue with their sessions when the Identity Server can't retrieve their passwords. If this option is not selected, users are denied access when their passwords can't be retrieved.

   Retain Previous Principal: Select this option to retain the principal obtained from the previous authentication method. If you do not select this option, then the principal will be used from the method associated with this class.

   Password to be retrieved: If your users have been configured to use a universal password, select **Universal Password**. Otherwise, select **Simple Password**.

   NOTE

   - Set the Universal Password Retrieval options in the configuration of the Universal Password policy, so that the policy allows the password to be retrieved from the User Store.
   - User must reset the password after configuring the password policy for universal password.

   For more information about Unable to retrieve Universal Password from eDirectory using PasswordFetchClass issue, see TID 7007114.

   The user object must be looked up and found in an eDirectory user store for retrieval to succeed. This is done by matching the currently authenticated user using the CN attribute. If your CN does not match in both of your directories (common if using Active Directory and eDirectory), then the DN of the user can be used to locate the matching user object. When NetIQ Identity Manager is used between Active Directory and eDirectory, you will find the Active Directory DN value populated in the DirXML-ADContext attribute which can be used for lookup / matching. If no attribute has the DN value populated, then the Auto Provision feature can be used.
4. Configure the following UserStore Lookup Settings:

   Based on the CN of the user object: The CN users are mapped between two different user stores. CN is mapped with for retrieving the password from user store. For Example - Active Directory CN is mapped with eDirectory CN for retrieving the password from eDirectory user store.

   Based on the Attribute value of the user object: The user names are detected and handled in LDAP attribute or DN users of the Active Directory are mapped with LDAP attribute of the eDirectory. If you select this option, then specify the attribute value in attribute details of the **Attribute name of DN** and enable the **Auto Provision** check box if required.

Attribute Name of the DN: Specify the attribute name of the DN.

This attribute must contain the CN of user whose password you want to obtain. For example, if you are trying to obtain a password from eDirectory for a user with cn=a,dc=b, then you need to specify name of the attribute, which value is cn=a,dc=b.The passwordfetchclass tries fetching the password from the current user store based on the value of the LDAP attribute specified, which are mapped to user's DN of the Active Directory.

Auto Provision: If you enable this check box, then the passwordfetchclass tries fetching the password from LDAP attribute specified above which has the value of the DN users of the Active Directory and retrieves the password, else it prompts to log in to the eDirectory. If the log in is successful, then the LDAP attribute value gets populated with the DN user of the Active Directory. When the user is logged next time the same value is used.

5 Click **OK**.

6 Create a method for this class.

For instructions, see “Configuring Authentication Methods” on page 287.

7 Assign the password fetch method as the second method for a contract that is using one of the following for its authentication method:

- RADIUS. See “RADIUS Authentication” on page 310.
- X.509. See “Mutual SSL (X.509) Authentication” on page 318.
- OpenID. See “OpenID Authentication” on page 324.
- Smart Card. See “Configuring Access Manager for NESCM” on page 327.
- Kerberos. See “Kerberos Authentication” on page 331.
- Time-Based One-Time Password. See “Two-Factor Authentication Using Time-Based One-Time Password (TOTP)” on page 305

8 Click **Apply** and update the Identity Server.

Configuring Access Manager for NESCM

To use a smart card with Access Manager, you need to configure Access Manager to use the eDirectory server where you have installed the Novell Enhanced Smart Card Login Method for NMAS (NESCM). You then need to create a contract that knows how to prompt the user for the smart card credentials. The last task is to assign this contract to the protected resources that you want protected with a smart card. The following sections describe the prerequisites and the tasks:

- “Prerequisites” on page 327
- “Creating a User Store” on page 328
- “Creating a Contract for the Smart Card” on page 329
- “Assigning the NESCM Contract to a Protected Resource” on page 330
- “Verifying the User’s Experience” on page 331
- “Troubleshooting” on page 331

Prerequisites

❒ Ensure that you can authenticate to the eDirectory server by using the smart card from a workstation.

- The NESCM method needs to be installed on the eDirectory server and the workstation. See "Installing the Method" (http://www.novell.com/documentation/iasclient30x/nescm_install/data/b7gx5la.html) in the *Novell Enhanced Smart Card Method Installation and Administration Guide* (http://www.novell.com/documentation/iasclient30x/nescm_install/data/bookinfo.html).
- The NESCM method needs to be configured. See "Configuring the Server" (http://www.novell.com/documentation/iasclient30x/nescm_install/data/b7tf2gi.html) in the *Novell Enhanced Smart Card Method Installation and Administration Guide* (http://www.novell.com/documentation/iasclient30x/nescm_install/data/bookinfo.html).
- Provision your smart card according to your company policy.

❒ Ensure that you have a basic Access Gateway configuration with a protected resource that you want to protect with a smart card. For more information, see Installing Access Manager Appliance in the NetIQ Access Manager Appliance 4.2 Installation and Upgrade Guide.

Creating a User Store

The Identity Server must be configured to use the eDirectory replica where you have installed the NESCM server method.

- If you have already configured the Identity Server to use this replica, skip this section and continue with "Creating a Contract for the Smart Card" on page 329.
- If your Identity Server is using a different user store, you need to configure the Identity Server.

To configure the Identity Server for the eDirectory replica that has the NESCM method:

1. In the Administration Console, click **Devices** > **Identity Servers** > **Edit** > **Local**> **User Stores** > **New**.
2. On the *Create User Store* page, fill the following fields:

   Name: A display name for the eDirectory replica (for example, `nescm_replica`).

   Admin Name: The distinguished name of the admin user of the directory. Administrator-level rights are required for setting up a user store.

   Admin Password and Confirm Password: The password for the admin user and the confirmation for the password.

   NOTE: If the admin account's password needs to be changed in the LDAP directory due to some issue, then change the admin password in the Create User Store page accordingly and apply the change. Else, this admin account of the user store will get locked.

   Directory Type: Select eDirectory.
3. In the **Server replica** section, click **New**, and fill the following fields:

   Name: The display name for the LDAP directory server (for example, `nescm_server`).

   IP Address: The IP address of the LDAP directory server. The port is set automatically to the standard LDAP ports.
4. Click **Use secure LDAP connections**. You must enable SSL between the user store and the Identity Server. The port changes to 636, which is the secure LDAP port.
5. Click **Auto import trusted root**.

6. Click **OK** to confirm the import.
7. Select the **Root CA Certificate** to trust any certificate signed by that certificate authority.
8. Specify an alias, then click **OK**.

   An alias is a name you use to identify the certificate used by Access Manager.
9. Click **Close**, then click **OK**.
10. Under **Server Replicas**, verify the **Validation Status**.

    The system displays a green check mark if the connection is valid.
11. Set up a search context.
12. Click **Finish** to save the information.
13. Continue with "Creating a Contract for the Smart Card" on page 329.

Creating a Contract for the Smart Card

You need to create a contract that uses the NESCM method. To do this, you need to first create an NMAS class, then a method that uses that class. The last task is to create a contract that uses the method. The following sections describe these tasks:

- "Creating an NMAS Class for NESCM" on page 329
- "Creating a Method to Use the NMAS Class" on page 329
- "Creating an Authentication Contract to Use the Method" on page 330

Creating an NMAS Class for NESCM

When you create a class, you can specify values for properties. In the following steps, you specify a property value that determines the sequence of login prompts that the user receives when authenticating with a smart card.

1. In the Administration Console, click **Devices** > **Identity Servers** > **Edit** > **Local** > **Classes** > **New**.
2. Specify a display name for the class (for example, `Class-NMAS-NESCM`).
3. For the **Java class**, select **NMASAuthClass** from the selection list.
4. Click **Next**.
5. On the *Specify Properties* page, click **New**.
6. Specify the following values for the property:

   Property Name: Specify `NMAS_LOGIN_SEQUENCE`

   Property Value: Specify `Enhanced Smart Card`

   The Property Value matches the method name as displayed in the **NMAS** task > **NMAS Login Methods**.
7. Click **OK**, then click **Finish**.
8. Continue with "Creating a Method to Use the NMAS Class" on page 329.

Creating a Method to Use the NMAS Class

When you create a method, you can specify property values that are applied to just this method and not the entire class. In this tutorial, we want the method to use the same login sequence as the class. The method also allows you to specify which user stores can use the method. For a smart card method, you need to ensure that the user store or stores specified for the method have NESCM installed.

1. On the Local page for the Identity Server, click **Methods** > **New**.
2. Specify a **Display name** (for example, `Method-NMAS-NESCM`).
3. From the **Class** selection list, select the class created in "Creating an NMAS Class for NESCM" on page 329.
4. In the **Available user stores list**, select the user store created in "Creating a User Store" on page 328, then click the left-arrow to move this user store into the **User stores** list.

   Leave other settings on this page unchanged.
5. Click **Finish**.
6. Continue with "Creating an Authentication Contract to Use the Method" on page 330.

Creating an Authentication Contract to Use the Method

Contracts are the element you can assign to a protect a resource.

1. On the Local page for the Identity Server, click **Contracts** > **New**.
2. Specify a **Display name** (for example, `Contract-NMAS-NESCM-UserStore1`).
3. Enter a **URI** (for example, `nescm/test/uri`).

   The URI is used to identify this contract for external providers and is a unique path value that you create.
4. In the **Available methods** list, select the method created in "Creating a Method to Use the NMAS Class" on page 329, then click the left-arrow to move this method into the **Methods** list.

   All other fields can remain in the default state.
5. (Conditional) If you want the user's credentials (username and password) to be available for Identity Injection policies, add the password fetch method as a second method for the contract.

   For more information about this method and class, see "Password Retrieval" on page 325.
6. Click **Next**, then configure a card for the contract by filling in the following fields:

   ID: (Optional) Specify an alphanumeric value that identifies the card. If you need to reference this card outside of the Administration Console, you need to specify a value here. If you do not assign a value, the Identity Server creates one for its internal use.

   Text: Specify the text that is displayed on the card to the user, for example Smart Card.

   Image: Select the image to display on the card. You can select the NMAS Biometrics image or you can select the **Select local image** option and upload an image that your users can associate with using this smart card authentication contract.

   Show Card: Determine whether the card is shown to the user, which allows the user to select and use the card for authentication. If this option is not selected, the card is only used when a service provider makes a request for the card.
7. Click **Finish**, then click **OK**.
8. Update the Identity Server.
9. Update the Access Gateway.
10. Continue with "Assigning the NESCM Contract to a Protected Resource" on page 330

Assigning the NESCM Contract to a Protected Resource

Contracts must be created before they can be assigned to protected resources. The following steps explain how to assign the NESCM contract to an existing protected resource. If you have not created a protected resource, see “Configuring Protected Resources” on page 101.

1. In the Administration Console, click **Devices** > **Access Gateways** > **Edit** > **[Name of Reverse Proxy]**.

   The reverse proxy must be configured with a resource that you want to protect with the smart card.
2. Click the **Protected Resource** link for the proxy service where you want to assign the NESCM contract.
3. To enable the NESCM contract on an existing protected resource, click the **Authentication Procedure** link for that resource, then select the NESCM contract created in “Creating an Authentication Contract to Use the Method” on page 330.

   If the contract is not listed, ensure that you have updated the changes to the servers, first to the Identity Server and then the Access Gateway. If you have multiple Identity Server configurations, ensure that the Access Gateway is assigned to the Identity Server configuration that contains the NESCM contract (click **Access Gateways** > **Edit** > **Reverse Proxy / Authentication**).
4. Click **OK**.
5. Click the **Access Gateways** task, then update the Access Gateway.
6. Continue with “Verifying the User’s Experience” on page 331.

Verifying the User’s Experience

1. From the smart-card-equipped workstation, browse to and select the URL of the proxy service where the protected resource requiring NESCM type authentication is enabled.
2. When prompted by Access Manager, enter a **username**.
3. When prompted for the smart card password, enter a password (the smart card PIN).

If the Smart Card contains a certificate that meets the defined criteria (in this example, a matching Subject name and trusted signing CA), the user is now successfully authenticated to the IDP and is connected through the Access Gateway to the protected resource.

Troubleshooting

Error	Resolution
Authentication fails without prompting the user for the token	Verify that you have configured the class and method correctly. See “Creating an NMAS Class for NESCM” on page 329 and “Creating a Method to Use the NMAS Class” on page 329.
Certificate validation fails	Verify that a trusted root object created for the signing CA of the certificate on the smart card exists in the eDirectory trusted root container.

Kerberos Authentication

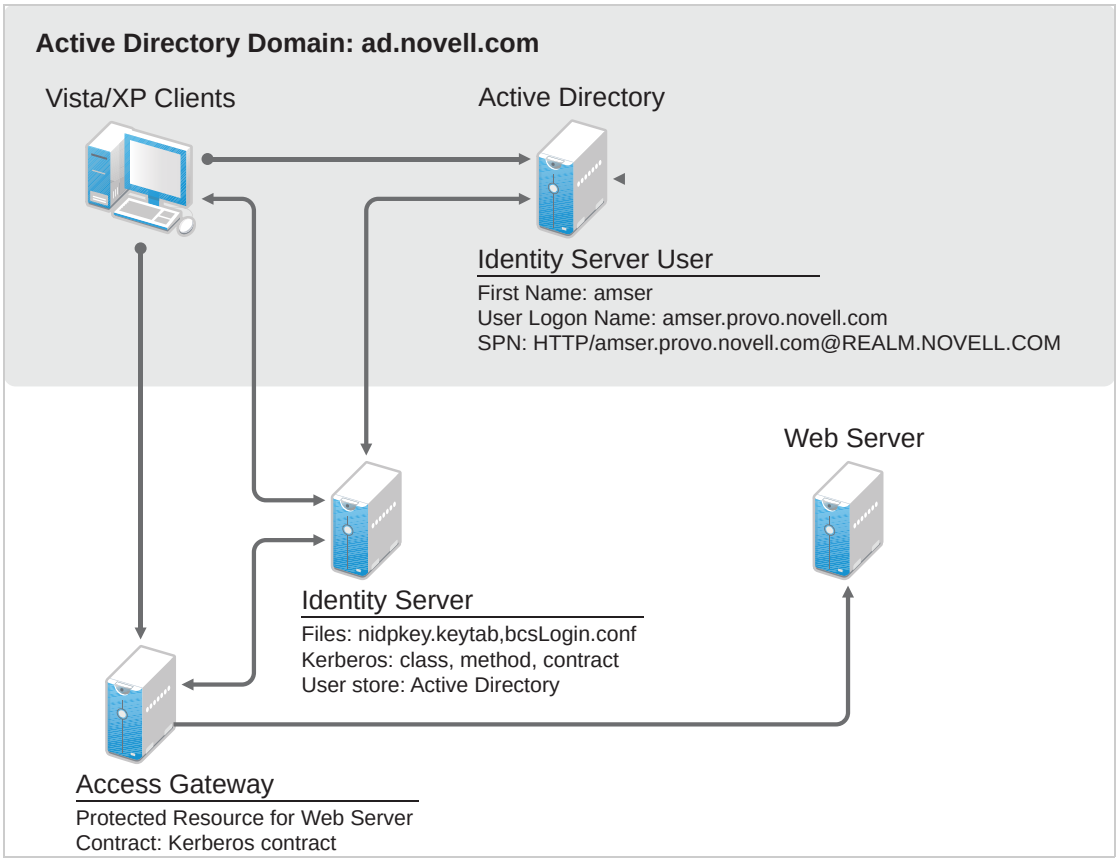
Kerberos is an authentication method that allows users to log in to an Active Directory domain. This authentication method provides them with a token, which an Identity Server can be configured to use as a contract. This provides single sign-on for the user between Active Directory and the Identity Server.

Kerberos authentication is achieved using SPNEGO with GSS-API (JGSS). SPNEGO (RFC 2478 - Simple and Protected GSSAPI Negotiation implementation in Microsoft Windows 2000/XP/2k3/2k8) is a GSSAPI mechanism for extending a Kerberos single-sign-on environment to Web transactions and services. It lets peers determine which GSSAPI mechanisms are shared and lets them select one and establish a security context with it. SPNEGO's most visible use is in Microsoft's HTTP Negotiate authentication mechanism.

The Kerberos module for Access Manager is implemented as additional out-of-the-box authentication mechanism to securely negotiate and authenticate HTTP requests for protected resources. This makes it possible to seamlessly authenticate (single-sign-on) to the Identity Server from enterprise-wide Microsoft Windows Domain Logon.

This section explains how to configure Active Directory, the Identity Server, and the Access Gateway for Kerberos authentication to a protected Web server.

Figure 5-4 *Example Kerberos Configuration*



Kerberos requires the following configuration tasks:

- “Prerequisites” on page 332
- “Configuring Active Directory” on page 333

- “Configuring the Identity Server” on page 335
- “Configuring the Clients” on page 342
- “Configuring the Access Gateway for Kerberos Authentication” on page 343

Prerequisites

Kerberos authentication is supported for the following configuration:

- Clients must be running one of the following operating systems:

  Windows XP with Internet Explorer 7 or later. Some minimal testing has been done with Internet Explorer 6. To make Kerberos work with Internet Explorer 6, you need to enable integrated Windows authentication. For information about how to enable this feature, see “Authentication Uses NTLM instead of Kerberos” (http://technet.microsoft.com/en-us/library/cc779070.aspx).

  Windows Vista with the latest version of Internet Explorer.

  Windows 7 with Internet Explorer 8 or later. Be aware of the following issues:

  - Internet Explorer needs to have the Internet Options configured to trust the URL of the Identity Server.
  - The keytab file must be configured to trust more than DES encryption. If you created your keytab file for an earlier version of Access Manager where only DES was supported, you need to recreate the keytab file. For the new procedure, see “Configuring the Keytab File” on page 334.

  For more information about these issues, see TID 7006036 (http://www.novell.com/support/viewContent.do?externalId=7006036&sliceId=1).
- Active Directory must be configured to contain entries for both users and their machines. Active Directory must be running on Windows Server 2003 Enterprise SP2, Windows Server 2008, Windows Server 2008 R2, or Windows Server 2012.
- Active Directory and the Identity Server must be configured to use a Network Time Protocol server. If time is not synchronized, authentication fails.
- If a firewall separates the Active Directory Server from the Identity Server, the firewall needs to open ports TCP 88 and UDP 88 so that the Identity Server can communicate with the KDC on the Active Directory Server.

Configuring Active Directory

You must create a new user in Active Directory for the Identity Server, set up this user account to be a service principal, create a keytab file, and add the Identity Server to the Forward Lookup Zone. These tasks are described in the following sections:

- “Installing the spn and the ktpass Utilities for Windows Server 2003” on page 333
- “Creating and Configuring the User Account for the Identity Server” on page 333
- “Configuring the Keytab File” on page 334
- “Adding the Identity Server to the Forward Lookup Zone” on page 335

Installing the spn and the ktpass Utilities for Windows Server 2003

When you install Windows Server 2003 and Active Directory, the spn and ktpass utilities are not installed in a default installation. These utilities are installed in a default Windows Server 2012 installation.

You need the spn and ktpass utilities to configure the Identity Server for Kerberos authentication.

1. Insert the Windows 2003 CD into the CD drive.
2. To install the utilities, run `\SUPPORT\TOOLS\SUPTOOLS.MSI` on the CD.

   The utilities are installed in `C:\Program Files\Support Tools`.

Creating and Configuring the User Account for the Identity Server

1. In **Manage Your Server** on your Windows server, select the **Manage users and computers in Active Directory** option.
2. Select to create a new user.
3. Fill in the following fields:

   First name: Specify the hostname of the Identity Server. This is the username. For the example configuration, this is `amser`.

   User logon name: Specify `HTTP/<Identity_Server_Base_URL>`. For this example configuration, your Identity Server has a base URL of `amser.provo.novell.com`, and you would specify the following for the **User Logon Name**:

   ```
   HTTP/amser.provo.novell.com
   ```

   The realm is displayed next to the **User logon name**.

   User logon name (pre Windows 2000): Specify the hostname of the Identity Server. The default value must be modified. For the example configuration, this is `amser`.
4. Click **Next**, and configure the password and its options:

   Password: Specify a password for this user

   Confirm password: Enter the same password.

   User must change password at next logon: Deselect this option.

   Password never expires: Select this option.
5. Click **Next**, then click **Finish**.

   This creates the Identity Server user. You need to remember the values you assigned to this user for **First name** and **User logon name**.
6. To set the servicePrincipalName (spn) attribute for this user, open a command window and enter the following commands:

   ```
   setspn -A HTTP/<userLogonName> <userName>
   ```

   For this configuration example, you would enter the following command:

   ```
   setspn -A HTTP/amser.provo.novell.com@AD.NOVELL.COM amser
   ```

   This adds the servicePrincipalName attribute to the user specified with the value specified in the `-A` parameter.

   NOTE: For Domain Services for Windows, set HOST spn also by using this command: `setspn -A HOST/<userLogonName> <userName>`

   ---
7. (Optional) Verify that the user has the required servicePrincipalName attribute with a valid value. Enter the following command:

```
setspn -L <userName>
```

For this configuration example, you would enter the following command:

```
setspn -L amser
```

Configuring the Keytab File

The keytab file contains the secret encryption key that is used to decrypt the Kerberos ticket. You need to generate the keytab file and copy it to the Identity Server.

1 On the Active Directory server, open a command window and enter a `ktpass` command with the following parameters:

```
ktpass /out value /princ value /mapuser value /pass value
```

The command parameters require the following values:

Parameter	Value	Description
/out	<*outputFilename*>	Specify a name for the file, with `.keytab` as the extension. For example: `nidpkey.keytab`
/princ	<*servicePrincipalName*> @<*KERBEROS_REALM*>	Specify the service principal name for the Identity Server, then @, followed by the Kerberos realm. The default value for the Kerberos realm is the Active Directory domain name in all capitals. The Kerberos realm value is case sensitive.
/mapuser	<*identityServerUser*>@<*AD_DOMAIN*>	Specify the username of the Identity Server user and the Active Directory domain to which the user belongs.
/pass	<userPassword>	Specify the password for this user.

For this configuration example, you would enter the following command to create a keytab file named `nidpkey`:

```
ktpass /out nidpkey.keytab /princ HTTP/amser.provo.novell.com@AD.
NOVELL.COM /mapuser amser@AD.NOVELL.COM /pass novell
```

2 Copy the file to the default location on the Identity Server:

```
/opt/novell/java/jre/lib/security
```

3 If the cluster contains multiple Identity Servers, copy the keytab file to each member of the cluster.

Adding the Identity Server to the Forward Lookup Zone

1 In **Manage Your Server** on your Windows server, click **Manage this DNS server**.

2 Click **Forward Lookup Zone**.

3 Click the Active Directory domain.

4 In the right pane, right click, and select **New Host (A)**.

5 Fill in the following fields:

Name: Specify the hostname of the Identity Server.

IP Address: Specify the IP address of the Identity Server.

6 Click **Add Host**.

Configuring the Identity Server

You need to configure the Identity Server to use the Active Directory server as a user store, configure a Kerberos authentication class, method, and contract, create a configuration file, enable logging to verify the configuration, then restart Tomcat. These instructions assume that you have installed and configured an Identity Server cluster configuration.See Installing Access Manager Appliance in the NetIQ Access Manager Appliance 4.2 Installation and Upgrade Guide and “Identity Servers Cluster” on page 52.

- “Enabling Logging for Kerberos Transactions” on page 336
- “Configuring the Identity Server for Active Directory” on page 336
- “Creating the Authentication Class, Method, and Contract” on page 337
- “Creating the bcsLogin Configuration File” on page 338
- “Verifying the Kerberos Configuration” on page 339
- “(Optional) Using the Name/Password Form Authentication” on page 340
- “(Optional) Configuring the Fall Back Authentication Class” on page 341

Enabling Logging for Kerberos Transactions

Enabling logging is highly recommended. If Kerberos authentication does not function after you have finished the configuration tasks, the first step in solving the problem is to look at the `catalina.out`.

1. In the Administration Console, click **Devices** > **Identity Servers** > **Edit** > **Logging**.
2. Enable the **File Logging** and **Echo To Console** options.
3. In the **Component File Logger Levels** section, set **Application** to **debug**.
4. Click **OK**, then update the Identity Server.

Configuring the Identity Server for Active Directory

You need to either configure your Identity Server to use Active Directory as a user store or verify your existing configuration for your Active Directory user store.

1. In the Administration Console, click **Devices** > **Identity Servers** > **Edit**.
2. Click **Local**.
3. View your installed user stores.

   If you have already configured your Identity Server to use the Active Directory server, click its name.

   If you haven’t configured a user store for the Active Directory server, click **New**.
4. For a new user store, fill in the following fields. For an existing Active Directory user store, verify the values.

   Name: Specify a name of the user store for reference.

   Admin name: Specify the name of the administrator of the Active Directory server. Administrator-level rights are required for setting up a user store. This ensures read/write access to all objects used by Access Manager.

   Admin password and Confirm password: Specify the password for the administrator of the Active Directory server and confirm the password.

   Directory Type: Select **Active Directory**.

Search Contexts: For a new user store, click **New** and specify the context of the administrator of the Active Directory server. For an existing user store, verify that you have an entry for the context of the administrator and add one if it is missing.

5 (Conditional) For a new Active Directory user store, add a replica. In the **Server replicas** section, click **New**.

5a Fill in the following fields:

Name: Specify a name of the replica for reference. This can be the name of your Active Directory server.

IP Address: Specify the IP address of the Active Directory server and the port you want the Identity Server to use when communicating with the Active Directory server.

5b Configure the other fields to fit your security model.

5c Click **OK**.

6 (Optional) Specify values for the other configuration options.

7 To save your changes, click **OK** or **Finish**.

8 Continue with "Creating the Authentication Class, Method, and Contract" on page 337.

Creating the Authentication Class, Method, and Contract

1 In the Local page, click **Classes** > **New**.

2 Fill in the following fields:

Display name: Specify a name that you can use to identify this class.

Java class: Select **KerberosClass**.

The **Java class path** field displays the name of the KerberosClass.

3 Click **Next.**

4 Fill in the following fields:

Service Principal Name (SPN): Specify the value of the servicePrincipalName attribute of the Identity Server user. For this example configuration, this is `HTTP/amser.provo.novell.com`.

Kerberos Realm: Specify the name of the Kerberos realm. The default value for this realm is the domain name of the Active Directory server, entered in all capitals. The value in this field is case sensitive. For this example configuration, this is `AD.NOVELL.COM`.

JAAS config file for Kerberos: Verify the default path. This must be the same path to which you copied the keytab file (see Step 2 in "Configuring the Keytab File" on page 334) and end with the name of the configuration file, `bcsLogin.conf`.

Instructions for creating this file are in "Creating the bcsLogin Configuration File" on page 338.

Kerberos KDC: Specify the IP address of the Key Distribution Center. If multiple KDCs are present for fail-over support, then specify the IP addresses separated by colon (:). Maximum of 4 IP addresses can be configured.

If a L4 switch is configured for load balancing between the KDCs, then enter the virtual IP address of the L4 switch in this field.

User Attribute: Specify the name of the Active Directory attribute that combines the cn of the user with the DNS domain name to form its value. It is an alternate name for user login. Accept the default value unless you have set up a different attribute.

5 (Conditional) If you have configured your users to have multiple User Principal Names (UPN) so they can log in using different names (such as jdoe@abc.com, jdoe@bcd.com, and jdoe@cde.com), click **New**, specify the suffix (such as @abc.com), then click **OK**.

6 Click **Finish**.

IMPORTANT: You must create only one Kerberos class. This is caused by a limitation in the underlying Sun JGSS.

7 On the Local page, click **Methods** > **New.**

8 Fill in the following fields:

Display name: Specify a name that you can use to identify this method.

Class: Select the class that you created for Kerberos.

User stores: Move the Active Directory user store to the list of User stores. If you have only one installed user store, **<Default User Store>** can be used. If you have multiple user stores, the Active Directory user store must be in this list (or if it is configured to be the default user store, **<Default User Store>** must be in this list).

NOTE: The testing procedure to verify Kerberos authentication is dependent upon having the Active Directory user store configured as the default user store. See Step 13.

You do not need to configure properties for this method.

9 Click **Finish**.

10 In the Local page, click **Contracts** > **New.**

11 Fill in the following fields:

Display name: Specify a name that you can use to identify this method.

URI: Specify a value that uniquely identifies the contract from all other contracts.

Methods: From the list of **Available methods**, move your Kerberos method to the **Methods** list.

You do not need to configure the other contract options.

12 Click **Finish**.

13 (Optional) To use the procedure that verifies the authentication configuration, you need to make the Active Directory user store the default user store. In the Local page, click **Defaults**.

13a Fill in the following fields:

User Store: Select the name of your Active Directory user store.

Authentication Contract: Select the name of your Kerberos contract.

13b Click **OK**.

This allows you to log in directly to the Identity Server by using the Kerberos contract. If you have already logged in to the Active Directory domain on the Windows machine, single sign-on is enabled and you are not prompted to log in to the Identity Server.

14 On the Identity Servers page, click **Update**.

Wait until the Health icon turns green. Click **Refresh** to update the page.

15 If you have Access Gateways or that you want to configure to use the Kerberos contract, update these devices so that the Kerberos contract is available.

16 Continue with “Creating the bcsLogin Configuration File” on page 338.

Creating the bcsLogin Configuration File

If you are upgrading from 3.1 SP2 to 3.1 SP3, the syntax of the `bcsLogin.conf` file has changed.

To create the file:

1. Open a text editor. A sample bcsLogin.con file called `bcsLogin.conf.template` is included that can be edited. Open this file.
2. Enter the following lines. The file cannot contain any white space, only end-of-line characters. Two lines (principal and keyTab) need to specify unique information for your configuration. The principal line needs to specify the service principle name for the Identity Server. The keyTab line needs to specify the location of the keytab file. The following file uses the values of the example configuration for the principal and keyTab lines. The keyTab and ticketCache lines use the default path for SUSE Linux Enterprise Server (SLES).

   ```
   com.sun.security.jgss.accept {
   com.sun.security.auth.module.Krb5LoginModule required
   debug="true"
   useTicketCache="true"
   ticketCache="/opt/novell/java/jre/lib/security/spnegoTicket.cache"
   doNotPrompt="true"
   principal="HTTP/amser.provo.novell.com@AD.NOVELL.COM"
   useKeyTab="true"
   keyTab="/opt/novell/java/jre/lib/security/nidpkey.keytab"
   storeKey="true";
   };
   ```

   The Identity Server will check the Kerberos server for each user transaction. When you set the isInitiator value to false (`isInitiator="false"`) in the `bcsLogin.conf` file after the `keyTab="/opt/novell/java/jre/lib/security/nidpkey.keytab"` line, the Identity Server does not communicate to the Kerberos server.

   Path of `bcsLogin.conf` on SLES and Red Hat is `/opt/novell/java/jre/lib/security/`.

   NOTE: Before setting the value to "false", it is recommended that you access the protected site via https and keytab file is secure.

   ---

3. Save this file with a name of `bcsLogin.conf`.
4. Copy this file to the location specified in the **JAAS config file for Kerberos** field of Step 4 in "Creating the Authentication Class, Method, and Contract" on page 337.
5. Ensure that the file permissions are set correctly. They must be set to 644.
6. Restart Identity Server:

   ```
   /etc/init.d/novell-idp restart
   ```

   Whenever you make changes to the `bcsLogin.conf` file, you need to restart Tomcat.
7. If the cluster contains multiple Identity Servers, copy the `bcsLogin.conf` file to each member of the cluster, then restart Tomcat on that member.

Verifying the Kerberos Configuration

To view `catalina.out`:

1. In the Administration Console, click **Auditing** > **General Logging**.
2. In the Identity Servers section, select the `catalina.out` file.
3. Download the file and open it in a text editor.

4 Search for Kerberos and verify that a subsequent line contains a `Commit Succeeded phrase`. For the configuration example, the lines look similar to the following:

```
principal's key obtained from the keytab
principal is HTTP/amser.provo.novell.com@AD.NOVELL.COM
Added server's keyKerberos Principal HTTP/
amser.provo.novell.com@AD.NOVELL.COMKey Version 3key EncryptionKey: keyType=3
keyBytes (hex dump)=0000: CB 0E 91 FB 7A 4C 64 FE

[Krb5LoginModule] added Krb5Principal HTTP/
amser.provo.novell.com@AD.NOVELL.COM to Subject
Commit Succeeded
```

5 If the file does not contain any lines similar to these, verify that you have enabled logging. See "Enabling Logging for Kerberos Transactions" on page 336.

6 If the commit did not succeed, search backward in the file and verify the following values:

- Service Principal Name
- Name of keytab file

For the example configuration, the file must contain lines with text similar to the following:

```
Principal is HTTP/amser.provo.novell.com
KeyTab is /usr/lib/java/jre/lib/security/nidpkey.keytab
```

7 (Conditional) If you make any modifications to the configuration, either in the Administration Console or to the bcsLogin file, restart Tomcat on the Identity Server.

(Optional) Using the Name/Password Form Authentication

You can configure the IP address or the range of IP addresses of the clients for which the kerberos authentication must be skipped or performed using the kerberos.exclude or kerberos.include keywords respectively.

NOTE: You can specify only `kerberos.exclude` or `kerberos.include` argument in the `kerb.properties` file not both.

To configure this option, add the following entry in the `kerb.properties` file:

- kerberos.exclude=IP Address/Range separated by comma.
- kerberos.include=IP Address/Range separated by comma.

For example:

```
kerberos.exclude=1.1.1.1-9.255.255.255,10.50.1.1 - 10.50.1.50,11.1.1.1-
255.255.255.255
```

or

```
kerberos.include=10.1.1.1-10.49.255.255,10.50.1.51-10.255.255.255
```

For the clients coming from the IP addresses specified in `kerberos.exclude`, Kerberos authentication will be skipped and will fall back to the custom authentication class. See "(Optional) Configuring the Fall Back Authentication Class" on page 341.

For the clients coming from the IP addresses that are not specified in `kerberos.include`, kerberos authentication will be skipped and will fall back to the custom authentication class. See "(Optional) Configuring the Fall Back Authentication Class" on page 341.

The `kerb.properties` file is available at :

/opt/novell/nam/idp/webapps/nidp/WEB-INF/classes/kerb.properties

(Optional) Configuring the Fall Back Authentication Class

You can configure an optional authentication class that has to be executed when either kerberos authentication fails or when kerberos authentication has to be skipped.

For information about how to skip the kerberos authentication for certain IP addresses, see "(Optional) Using the Name/Password Form Authentication" on page 340

To configure the fall back authentication class:

1. Go to the **Identity Server Cluster** > **Edit** > **Local** > **Methods** > **(Kerberos Method)** > **Properties** tab.
2. Add a new property `/value pair` with name as FALLBACK_AUTHCLASS and set the property value to be the qualified class name such as `com.novell.nidp.authentication.local.PasswordClass.`

   The class name value must be same as the value configured in the Java class path of the class at **IDP Cluster** > **Edit**> **Local** > **Classes**> **(Authentication class).**

NOTE: If your authentication class requires a custom JSP file for seeking credentials, add the property `JSP` and specify the name of the jsp file. When the JSP property is not specified, Identity Server will use the default login.jsp for seeking the credentials.

If you want to fall back to basic authentication, configure any one of the following properties:
`Property Name: FALLBACK_AUTHCLASS`

`Property Value: Basic or com.novell.nidp.authentication.local.BasicClass`

NOTE: The property name is case-sensitive.

For example, if you want to fall back to Radius, configure the following properties for the kerberos method:

FALLBACK_AUTHCLASS=com.novell.nidp.authentication.local.RadiusClass`JSP=radiusloginServer=<<radius IPs with comma separate>>SharedSecret=<<secret string>>Port=<<port>>ReplyTime=7000 (in milli seconds, this is optional)ResendTime=2000 (in milli seconds, this is optional)Retry=5 (this is optional)Password=false`

NOTE: The property name is case-sensitive.

Also, you can configure fall back to other mechanism based on the incoming header. In the kerberos Method, add the name/value in the property field with name as NO_NEGO_HEADER_NAME and in the value filed you can provide the header, which needs to be ignored for the kerberos authentication.

For Example, in the kerberos method properties, if you configure the name as NO_NEGO_HEADER_NAME with value X-NovINet. Then if the client comes with header X-NovINet, the kerberos class will not be executed and it will fall back to the name password form by default or to the configured fall back mechanism.

Configuring the Clients

1. Add the computers of the users to the Active Directory domain.

For instructions, see your Active Directory documentation.

2 Log in to the Active Directory domain, rather than the machine.

3 (Conditional) If you are using Internet Explorer, configure the web browser to trust the Identity Server:

3a Click **Tools** > **Internet Options** > **Security** > **Local intranet** > **Sites** > **Advanced**.

3b In the **Add this website to the zone** text box, enter the Base URL for the Identity Server, then click **Add**.

In the configuration example, this is `http://amser.provo.novell.com`.

3c Click **Close** > **OK**.

3d Click **Tools** > **Internet Options** > **Advanced**.

3e In the Security section, select **Enable Integrated Windows Authentication**, then click **OK**.

3f Restart the browser.

4 (Conditional) If you are using Firefox, configure the Web browser to trust the Identity Server:

4a In the URL field, specify `about:config`.

4b In the **Filter** field, specify **network.n**.

4c Double click `network.negotiate-auth.trusted-uris`.

This preference lists the sites that are permitted to engage in SPNEGO Authentication with the browser. Specify a comma-delimited list of trusted domains or URLs.

For this example configuration, you would add `http://amser.provo.novell.com` to the list.

4d If the deployed SPNEGO solution is using the advanced Kerberos feature of Credential Delegation, double-click `network.negotiate-auth.delegation-uris`. This preference lists the sites for which the browser can delegate user to the server. Specify a comma-delimited list of trusted domains or URLs.

For this example configuration, you would add `http://amser.provo.novell.com` to the list.

4e Click **OK**, then restart your Firefox browser.

5 In the URL field, enter the base URL of the Identity Server with port and application. For this example configuration:

```
http://amser.provo.novell.com:8080/nidp
```

The Identity Server must authenticate the user without prompting the user for authentication information. If a problem occurs, check for the following configuration errors:

- Verify the default user store and contract. See Step 13.
- View the Identity Server logging file and verify the configuration. See “Verifying the Kerberos Configuration” on page 339.
- If you make any modifications to the configuration, either in the Administration Console or to the `bcsLogin` file, restart Tomcat on the Identity Server.

6 (Conditional) If you have users who are outside the firewall, they cannot use Kerberos. SPNEGO defaults first to NTLM, then to HTTPS basic authentication. Access Manager does not support NTLM, so the NTLM prompt for username and password fails. The user is then prompted for a username and password for HTTPS basic authentication, which succeeds if the credentials are valid.

To avoid these prompts, you can have your users select the **Automatic logon with current user name and password** option in Internet Explorer. To access this option, click **Tools** >**Internet Options** >**Security** >**Custom Level**, then scroll down to **User Authentication**.

Configuring the Access Gateway for Kerberos Authentication

If you have set up a Web server that you want to require Kerberos authentication for access, you can set up a protected resource for this Web server as you would for any other Web server, and select the name of your Kerberos contract for the authentication procedure. For instructions, see “Configuring Protected Resources” on page 101.

When using Kerberos for authentication, the LDAP credentials are not available. If you need LDAP credentials to provide single sign-on to some resources, see “Password Retrieval” on page 325.

Risk-based Authentication

This section describes risk-based authentication and how to configure it.

- “Overview” on page 343
- “How Risk-based Authentication Works” on page 344
- “Why Risk-based Authentication” on page 347
- “Features of Risk-based Authentication” on page 348
- “Key Terms” on page 352
- “Understanding Risk-based Authentication through Scenarios” on page 354
- “Understanding Risk Score Calculation” on page 360
- “Configuring Risk-based Authentication” on page 362
- “Configuring an External Database to Store User History” on page 362
- “Troubleshooting Risk Rule Configuration” on page 364

Overview

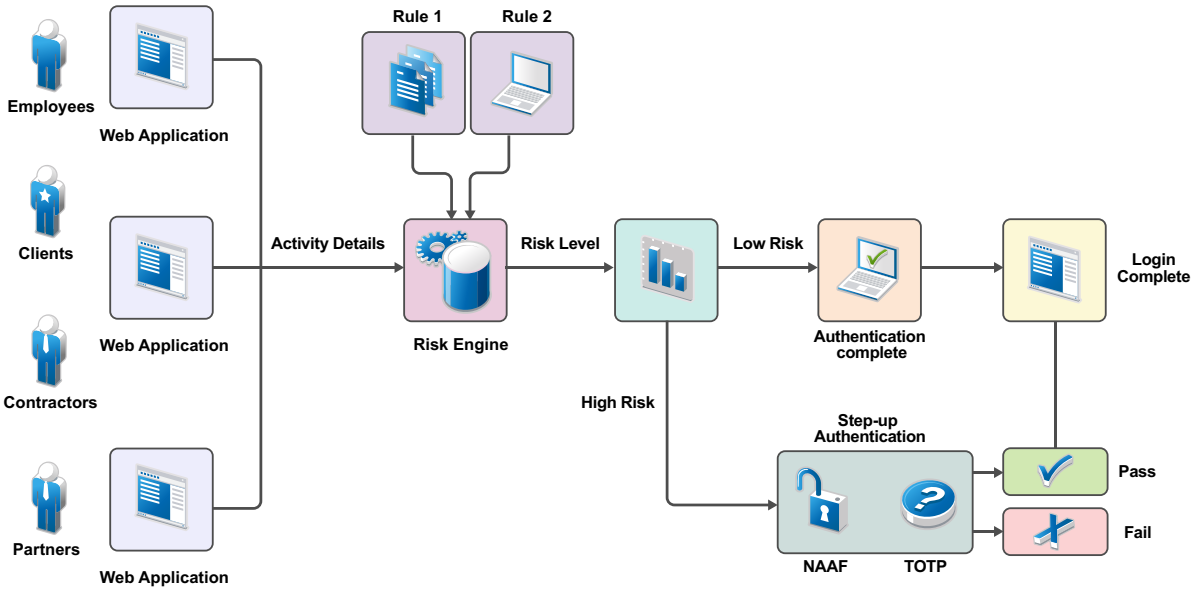
Traditional password-based authentication systems have their own limitations at implementing security in an organization. Enhancing the strength of the password is inadequate to prevent security threats. Thus, there is a need to explore and apply better authentication techniques such as risk-based authentication.

Risk-based authentication provides context-aware access control that acts to balance the level of trust against risk. It enables organizations to address access-related risks and improves user experience. Risk-based authentication enables to validate the risk of an access request at the run time and take appropriate actions such as forcing an advanced authentication or denying access.

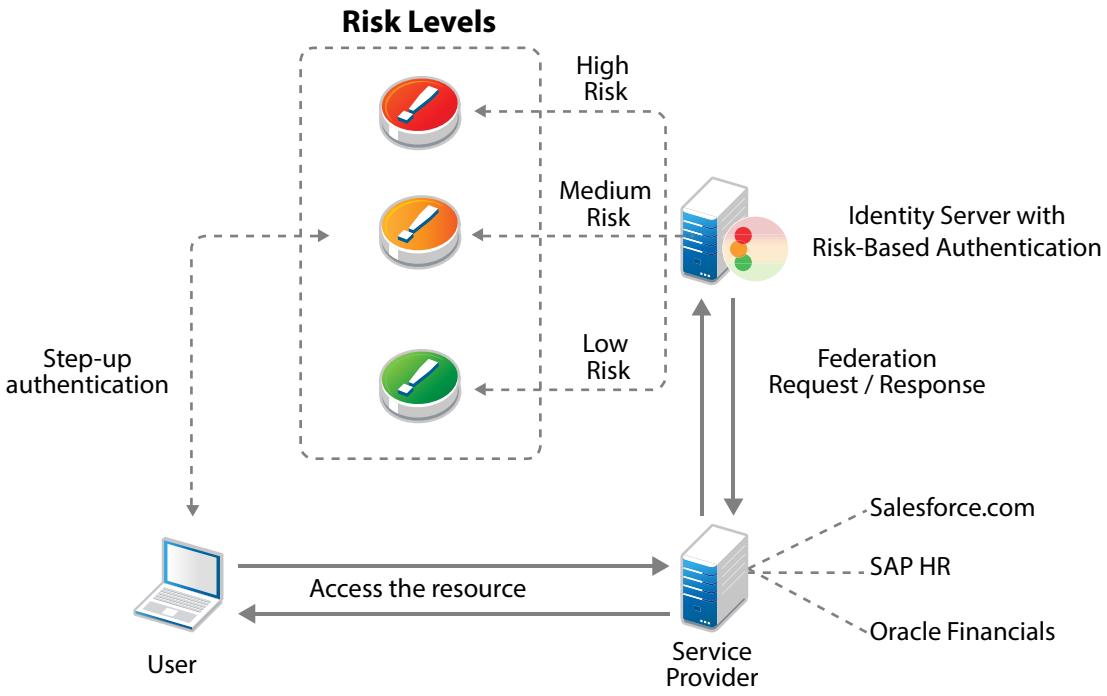
You can also assess risk in a federated setup with service providers such as Salesforce.com, SAP HR, and Oracle Financial with protocols such as SAML and WS Fed.

The Access Gateway can also control access for a protected resource based on the risk score.

The following illustration depicts risk-based authentication process:



The following illustration depicts the risk-based authentication in a federated setup:



How Risk-based Authentication Works

You can configure risk-based authentication in the following two ways:

- Risk assessment and risk mitigation before authenticating a login attempt
- Risk assessment and risk mitigation after authenticating a login attempt

Risk Assessment and Risk Mitigation before Authenticating a Login Attempt

You can assess the potential risk of a particular login attempt before authenticating the user and then mitigate the risk, if required. You can also configure specific authentication mechanisms based on the risk. In this scenario, user profile is not involved.In this scenario, risk-based authentication works by developing a risk score based on the following parameters:

- IP Address
- Cookie
- HTTP Header
- Geolocation
- External parameters
- Time of login
- Device ID (without user attributes)
- Custom rule (without user attributes)

This risk score is then evaluated against defined risk levels. You can define the risk levels based on the sensitivity of the information. After the risk level is identified, the authentication mechanism is selected and the user is authenticated. In cases of high risk, the user is either denied access or is required to go through additional authentication methods.

NOTE: You cannot record history in this configuration because there are no user-context. If you want to use history with pre-authentication risk assessment, you must configure a post risk-based authentication.

For example, an employee trying to log into a payroll application by using the corporate Intranet is authenticated through Kerberos authentication mechanism. However, the employee logging into the payroll application from outside the office must provide an x509 certificate for authentication.

The following graphic illustrates how risk-based authentication works in this scenario:



Risk Assessment and Risk Mitigation after Authenticating a Login Attempt

You can assess the potential risks associated with a particular login attempt after authenticating the user and then mitigate the risk, if required. In this scenario, risk-based authentication works by developing a risk score for each login attempt based on the following parameters:

- Geographical location
- Type of device
- IP address
- HTTP Header
- User attributes
- Cookie
- User last login
- Time of login
- External parameters
- Custom parameters

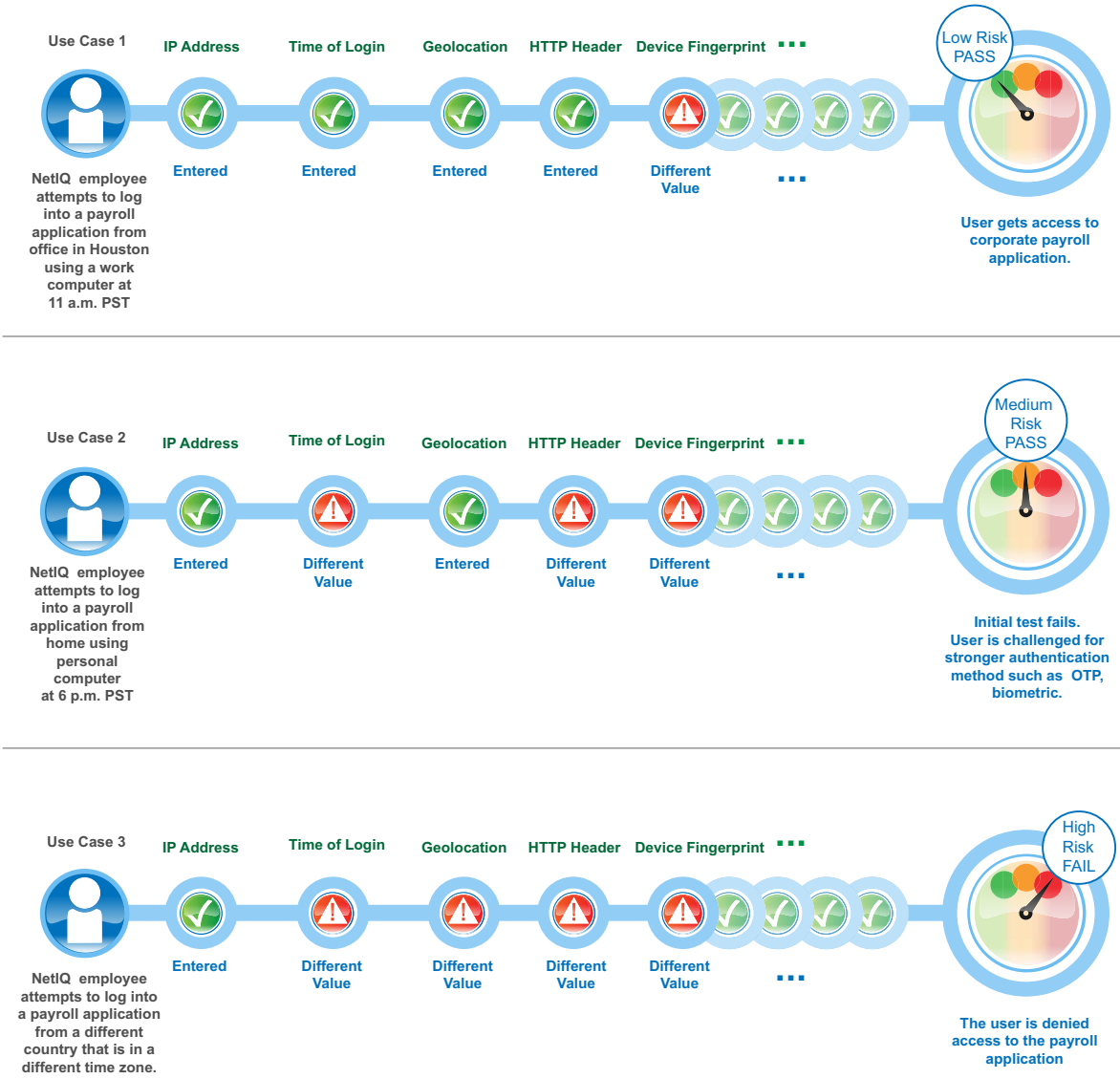
This risk score is then evaluated against defined risk levels. The risk levels are defined based on the sensitivity of the information. After the risk level is identified, the user is granted or denied access. In cases of high risk, the user is prompted for additional authentication to confirm the user identity one more time and assess the validity of the request.

For example: An employee logging into a payroll application by using the office laptop during the usual business hours from the same location and IP address will have a low risk score. Whereas, an attempt to access the payroll application by using a personal hand-held device from non-office location yields an elevated risk score. If the risk score for a user's access attempt exceeds the

defined risk score, the login attempt is considered as high risk, and the user might have to provide a higher level of authentication using a PIN or token. Additional authentication can be implemented by using techniques such as TOTP authentication or NetIQ Advance Authentication Framework (NAAF) methods. If the risk is too high, access can be denied.

NOTE: If you are configuring the step-up authentication with NAAF, ensure that you are on version 4.10 or 5.1.1. For more details about NAAF, see NAAF Documentation.

The following graphic illustrates how risk-based authentication works based on specific parameters:



Why Risk-based Authentication

Risk-based authentication helps you in achieving the following goals:

- Reduce fraud and the risk of improper access
- Enforce different levels of authentication depending on factors such as user activity and geolocation, and calculated risk score

- Improve user experience. Users need to provide additional details for authentication only when the associated risk prevails
- Access control in federated setups

Consider a scenario where a company named Company1 wants to protect its payroll application. Risk-based authentication enables Company1 to achieve the following actions:

- Restrict access to its contractual employees.
- Grant access to permanent employees during the company business hours between 9 a.m. to 5 p.m. After business hours, all employees must specify a one-time password in addition to login credentials.
- Grant special privileges to employees who work in the Finance department. For example, Company1 does not ask employees of the Finance department to specify a one-time password even if they log in after business hours.
- Grant access to the Self-Service tool along with the payroll application when contractual employees use Intranet to log in.
- Determine actions based on the priority of rule conditions. For example, type of employment is the most important criterion to grant access followed by the location of the user, and then the time of the login attempt.
- Grant access without any additional authentication if the user has successfully logged in within one month.
- Restrict access when an employee tries to log in from a specific geographical location.
- Grant or deny access based on the version of the web browser used for the login attempt.
- Deny access to any login attempt that originates from a handheld device.

Features of Risk-based Authentication

Risk-based authentication provides the following features:

- **Risk identifications by using rule definitions:** Risk-based authentication helps you define rules that are classified as follows.

Rule Category	Rule Description
IP Address	Use this rule to define a condition to track login attempts from an IP address, range of IP addresses, or an IP subnet range. For example: If you are aware that login attempts from a specific range of IP addresses are riskier in nature, you can define a rule to watch for such login attempts. When a request originates from the specified IP address range, you can prompt for additional authentication. You can also configure this rule to consider the list of IP addresses from an external provider for risk assessments. **IMPORTANT:** It is not possible to create a rule using the IP subnet condition. Instead you can use the IP address range condition to select a range of IP addresses in the rule. **NOTE:** The IP address may not be the clients IP address, but that of an intermediate device. In such a case, you need to use the X-Forwarded-For to determine the IP address of the workstation

Rule Category	Rule Description
Cookie	Use this rule if you want to track login attempts from a browser-based application that has a specific cookie value or name. For example: Consider a scenario where you have a financial application and a user accessing this application has cookies stored on the browser. If the cookie has a specific value or name, the user can be granted access without additional authentication. But, if the user accessing the application has no cookies stored, then you can request additional authentication to validate the user.
HTTP Header	Use this rule to track the HTTP header of requests based on the name or the value contained in the HTTP header. For example, if you want to track HTTP requests containing custom HTTP header information, you can define the action to be performed on evaluation of this rule.
User Last Login	This rule creates a cookie in browser after a successful step up authentication. Subsequent login verifies this cookie. Use this rule to define the duration for which the cookie is valid. On expiry of the specified duration, the user is prompted for additional authentication. For example, this rule can be used to evaluate if the user is logging in by using a device that was used earlier for a login attempt. You can define the risk level and also request additional authentication, as necessary.
User Profile	Use this rule to define a condition based on the LDAP attribute of the user. For example, you can define a rule to deny access if the employee ID of the user matches a specific string.
User Time of Login	Use this rule to define a condition based on the user's attempts to login at a specific time. For example, if the usual login pattern for an employee is between 9 a.m. to 5 p.m., you can define a rule that takes an action if the login pattern differs from the observed pattern.
Device ID	Use this rule to determine the type of device from which authentication was attempted. This is useful in situations where you want to track login attempts based on specific parameters, such as the user client or language. This rule creates a cookie in browser after a successful step up authentication. Subsequent login verifies this cookie and cookie is valid for duration configured in the Device ID rule. For example, an application supported on hand-held devices can be accessed through a variety of devices. In such a situation, you can define a rule that creates a cookie including details of the client from which the application is accessed. You can later use this data to introduce additional checks.
External Parameters	Use this rule to consider inputs from external providers to evaluate the risk associated with a login attempt. For example, if a user is already authenticated with an external authentication provider, Access Manager receives authentication details from that provider such as the method used for the authentication. Access Manager can use this information for evaluating the risk. **NOTE:** To use this rule, you must create a custom authentication class to retrieve details from an external provider. For more information, see "User Information Methods" and "Creating a Custom Rule Class" in the NetIQ Access Manager 4.2 Developer Guide.

Rule Category	Rule Description
Geolocation	Use this rule to track login attempts based on the geographical location of the user. You can track details ranging from a wide area such as a country or to a smaller area such as a region. For example, you can use this rule to introduce additional authentication attempts in the following scenarios: ◆ when a user logs in from a specific geolocation ◆ when a user accessing from a specific gelocation to be considered as a valid user and be granted access without further checks
Custom Rule	Lets you to define your own custom rules by using a custom Java authentication class. This is useful in deployments where the existing rules do not meet the requirements. For example: Consider a situation where the HTTP header contains the company name in an encoded format. You can create a custom rule to decode the HTTP header and retrieve the name of the company. This value can later be compared to an LDAP attribute, and based on the results, action to be taken. Consider a situation where a user logs in from India at a specific time each day. You can create a custom rule that will prevent login by the same user from USA within 24 hours of the previous login attempt. For more information about creating a custom class, see *NetIQ Access Manager 4.2 Developer Guide*.

- **Risk Engine:** The risk engine evaluates and assesses rules, including risk score and risk level processing. It ensures that the risk associated with the login attempt is considered during authentication.
- **History:** Each time a user makes a login attempt, the rules are evaluated and then based on the analysis, the user is either granted access or additional authentication is requested. These details are stored in an external database. The details recorded in history are:
  - Risk score after evaluation of the rule
  - Last login time of the user
  - Geolocation details such as city or country
  - IP address of the client
  - Risk level details after evaluation of the rule
  - Details of additional authentication
  - Details of error messages displayed during risk assessment
  - Time zone details

  It is recommended to configure Oracle, MYSQL, or Microsoft SQL Server as the database to store the risk-based authentication data. **Built-in Data Store** is not recommended in a production environment.
- **Risk Policies for risk mitigation:** A risk policy is a group of risk rules. A rule must be assigned to a risk policy for functioning. The risk policy was called `rule group` in Access Manager 4.1.
- **Authorization policies for protected resources:** You can define a condition group as part of the authorization policy that uses the risk score from the Identity Server to protect a resource. This provides an additional level of security for your environment. For more information, see "Configuring an Authorization Policy to Protect a Resource" on page 756.

- **Risk score and risk levels validation utility:** After configuring a risk policy and corresponding risk scores and actions, you can use the Validate utility to emulate the total risk score and the action in event of rule failure. The validation result indicates the total risk score and action. Based on these details, you can adjust the risk score and risk levels depending on your needs. For more information, see "Understanding How to Use the Validate Tool to Emulate Total Risk Score and Risk Levels" on page 364.
- **Risk Rule Validation Utility:** To determine the best way to implement risk levels and actions for your business needs, use the Risk Rule Validation Utility to test different risk rules and risk scores. For more information about this tool, see "Understanding How To Use the Risk Rule Validation Utility To Troubleshoot Rule Configuration" on page 366.
- **Cumulative Scoring:** You can configure to add the risk score of the current session while evaluating contracts.

  For example, assume that you have configured two contracts, Contract A and Contract B. `Contract A` contains both PreAuthRiskBasedAuthenticationClass and RiskBasedAuthClass (post authentication). In this case, cumulative scoring is enabled in RiskBasedAuthClass. When `Contract A` is executed, RiskBasedAuthClass considers the risk score calculated by PreAuthRiskBasedAuthenticationClass.

  `Contract A` is configured with the following two risk-based methods:

  - A_RiskMethod1, which uses a PreAuthRiskBasedAuthenticationClass that in turn uses a risk policy A_SamplePolicy1.
  - A_RiskMethod2, which uses a RiskBasedAuthClass that in turn uses a risk policy A_SamplePolicy2.

  For an arbitrary request, A_RiskMethod1 and A_RiskMethod2 calculate risk scores 100 and 150 respectively. As RiskBasedAuthClass has cumulative scoring enabled, total risk score calculated at the end of `Contract A` execution is 250.

  `Contract B` is configured with a method named B_RiskMethod. B_RiskMethod uses a RiskBasedAuthClass that in turn uses a risk policy named B_SamplePolicy. The risk score associated with this policy is 75. If evaluation of `Contract B` fails, 75 is returned as the risk score. If cumulative scoring is enabled, the final risk score of the session is 250 (risk score of Contract A) + 75 (risk score of Contract B) = 325. If cumulative scoring is not enabled, risk score is 75
- **Risk Score Reduction after a Successful Additional Authentication:** The **Reduction** option enables you to reduce the risk score by a specified value after a successful strong additional authentication.

  For example, Assume you want to reduce the risk score by 100 after a successful additional authentication. You have configured to prompt for an additional authentication when the risk score is more than 200. You have configured an authentication class with a risk policy that includes the following rules:

  - IPAddressRule: If this rule fails, the risk score is 125
  - HTTPHeaderRule: If this rule fails, the risk score is 150
  - UserProfileRule: If this rule fails, the risk score is 200

  Specify 100 in **Reduction**. For an arbitrary request, `IPAddressRule` and `HTTPHeaderRule` have failed and the effective risk score is 275. The user will be prompted for an additional authentication. If this authentication succeeds, the risk score becomes 175. If you have configured to share the risk score, the reduced risk score is shared. If you have enabled recording user history, the reduced value is logged.

  If the risk score value is less than one after reduction, risk score will be considered as zero.
- **Using External Parameters in Risk Assessment:** The **External Parameters Rule** enables you to consider inputs from external providers in evaluating the risk associated with a login attempt.

For example, if a user is already authenticated with an external authentication provider, Access Manager can receive authentication details such as the authentication type, IP address, and location of the user from that provider. Access Manager can then use this information for evaluating the risk.

You can configure to retrieve the input based on multiple parameters. Assume, you want to get the input if the user is from US and the method used to authenticate is Kerberos. Also, the user must not be using a device with the anti-virus software version 1.1.0.99. In this scenario, the configuration is as follows:

1. Create `Parameter Set 1`, set logical operator as `AND`, and configure the following two parameters:

Parameter Name	Regex	Operator	Parameter Value
Authentication Mechanism	NA	Is equal to	Kerberos
Location	NA	Is equal to	US

2. Create `Parameter Set 2` and configure the following parameter:

Parameter Name	Regex	Operator	Parameter Value
Anti-virus software	\\d+\\.\\d+\\.\\d+\\.\\d+	is not equal to	The version of anti-virus software is 1.1.0.99.

3. Set the logical operator between parameter sets as `AND`.

NOTE: To use this rule, you must create a custom class to retrieve details from an external provider. For more information, see “User Information Methods” and “Creating an Authentication Class” in the *NetIQ Access Manager 4.2 Developer Guide*.

Key Terms

This section explains key terms used in risk-based authentication.

Table 5-1 *Risk-based Authentication Terms*

Term	Description
Rule	A rule indicates a condition that you want to evaluate during a login attempt. For evaluation, a rule is linked to a risk policy. A rule can be assigned to multiple risk policy. For example, to assess the IP address of a user and the location from which the user logs in, you need two separate rules: One for IP address and another rule for location.
Risk Policy	You can combine one or more rules with a risk policy. A rule cannot be processed without being included in a risk policy. You can combine multiple rules in a risk policy.

Term	Description
Risk Score	The value that is returned if the rule conditions do not pass. For example, if you have set the risk score as 50, the risk score is displayed as 50 when the rule evaluation fails. It indicates that if the risk evaluation fails, 50 is the value returned to the risk engine. Assume that the IP address rule is assigned a risk score of 50 and the geolocation rule is assigned a risk score of 30. If both IP address rule and geolocation rule fail, the risk score is 80. If only the IP address rule fails, the risk score is 50. As the geolocation rule is evaluated successfully, the risk score is 0 and value of 30 is not stored in the risk engine.
Is/Is Not condition	When you configure a rule and select a parameter for assessment, you can determine how the conditions must match for each of the subparameters. For example, if you configure a rule to assess the IP address of a user, you can configure whether the IP address must be specific, be in a range, or be in a particular subnet. For example, if you want to assess whether the IP address of a user is within a range of 10.10.10.1 to 10.10.10.10, you can specify an **Is** condition in the rule configuration. This indicates that when the rule is evaluated, only IP addresses in the range of 10.10.10.1 to 10.10.10.10 must be considered as a valid IP addresses and then the user must be granted access. During the rule evaluation, if you want a rule to be passed when it does not meet a specific criteria, select **Is Not** in the rule configuration screen. For example, if you want to stop all login attempts from a particular IP address, then configure a rule using the **Is** condition. Using the same example as above, if you want to stop any login attempts from IP addresses in the range of 10.10.10.1 to 10.10.10.10, configure the rule using the **Is Not** condition.
Combination Rule	When you configure a set of rules, it is configured with the OR logical operator, by default. For example, if you have configured an IP address rule and a geolocation rule without any additional configuration, either the IP address rule is evaluated or the geolocation rule is evaluated. But, if you want both the IP address rule and the geolocation rule to be evaluated during a login attempt, configure a combination rule. A combination rule lets you use the **AND/OR** logical operators to configure a rule based on your preferences. For example, If you configure an IP address rule and a geolocation rule, select the **AND** operator to evaluate both rules. Whereas if you use the **OR** operator, either IP address rule or the geolocation rule is evaluated.
Risk Level	When a rule fails to evaluate successfully, the risk score is returned to the risk engine. If you have multiple rules configured, for each rule that fails to evaluate successfully, the risk score is added up to get a cumulative score. When configuring the risk level, you can determine the action the risk engine has to take if the total risk score crosses a certain limit and the risk level for the value. For example, you can determine that the risk is low if the total risk score is less than or equal to 50. Whereas if it is greater than 50, some action is required. Here action might mean an additional authentication request for the user.
Action	When a risk level and the associated risk score crosses the set threshold limit, you can configure the action as deny access or demand additional authentication. For example, if you have defined a risk level of High for a cumulative risk score of greater than 50, then you can specify that either the user must be denied access or additional authentication methods must be requested.

Understanding Risk-based Authentication through Scenarios

The following are few example configuration to describe how you can use risk-based authentication:

- Scenario 1
- Scenario 2
- Scenario 3
- Scenario 4
- Scenario 5

Scenario 1

You want to enable two-factor authentication for a user outside the corporate network, and single factor authentication when the user is in the corporate network.

You can configure risk-based authentication in this scenario by using `Risk-based Pre-Auth Class`. You can determine the authentication method required for an access based on the context before the user's identity is validated through authentication. You can choose from many authentication methods and apply them as per your security policy requirements.

For example, you can enable a simple username/password prompt when the user is in the internal network. If the user is trying to login from outside the corporate network using a known device, prompt for Advanced Authentication Framework (username/password and smartphone) authentication.

Configuration Steps:

1. Go to **Policies** > **Risk-based Policies** > **Risk Policy**.
2. Click the **Create Risk Policy** icon.
3. Under Add Risk Policy, specify the following details:

   Risk Policy Name: Specify a name.

   Policy Description: Specify the purpose of this policy.

   Assign Policy To: Select the Identity Server cluster and then configure an authentication class.

   - Select **Risk-based Pre-Authentication Class**.
   - Specify **Display Name**.
   - Click **Save**.
4. Create an IP Address rule.

   4a Under **Policy Rules**, click **Create Rule** and specify the following values:

   - **Rule Name**: Specify a name.
   - **Rule Definitions:** Select **IP Address Rule**.
   - Select **Allow if IP Address Is in the list**.
   - Specify the corporate IP Address details. For more information, see "IP Address" in Step 3 on page 750.

   4b Click **OK**.

   4c **If rule condition is met, then**: `Exit with Risk level as`.

   4d **Select Risk Level**: `Low`

   4e Click **OK**.

5 Create a Device ID rule.

5a Under **Policy Rules**, click **Create Rule** and specify the following values:

- **Rule Name**: Specify a name.
- **Rule Definitions:** Select **Device ID Rule**.
- Specify the details. For more information, see “Device ID” in Step 3 on page 750.

5b Click **OK**.

5c **If rule condition is met, then**: `Exit with Risk level as`.

5d **Select Risk Level**: `Medium`

5e **If rule condition is not met, add risk score:** `50`

5f Click **OK**.

6 Under Risk Levels, click **Actions** > **Add Risk Level** and create the following risk levels:

- **Low**

Field	Value
Risk Score	Less than 50
Risk Level	Low
Action	Authenticate by using Name/Password - Basic

- **Medium**

Field	Value
Risk Score	Equals to or greater than 50
Risk Level	Medium
Action	Authenticate by Advanced Authentication Framework (username/password and smartphone) **NOTE:** To enable this configuration, you must configure Advance Authentication Framework with Access Manager. For more information, see Access Manager - Advanced Authentication Plugin User Guide.

7 Click **OK**.

8 Create an authentication method. For more information, see “Configuring a Method for an Authentication Class” on page 749.

9 Create a contract. For more information, see “Configuring a Contract for an Authentication Class” on page 749.

10 Assign the contract to the protected resource.

Scenario 2

You want to configure an authentication mechanism or an additional authentication mechanism based on the type of the device.

You can configure risk-based authentication in this scenario by using `Risk-based Pre-Auth Class`. You can create a risk rule to choose an authentication mechanism or an additional authentication mechanism based on the type of device used by a user.

For example, if a user is logging in from a mobile device, you can prompt the user to provide an additional authentication such as SMS or One-Time Password based authentication after the user is authenticated. You can define an HTTP Header rule by using a user-agent property such as Mozilla/5.0 (iPhone; U; CPU iPhone OS 3_0) to verify whether the request is from a mobile.

Configuration Steps:

1. In the Administration Console, click **Risk-based Policies** > **Rules**.
2. Specify a name for the rule.
3. Select **HTTP Header Rule**.
4. Specify **HTTP Header Name** as `X-Forwarded-For`.
5. Select **Contains** in **HTTP Header Value** and specify `Mozilla/5.0 (iPhone; U; CPU iPhone OS 3_0)`.

   NOTE: You must configure NAT settings for this rule to work. See "Configuring NAT Settings" on page 755.
6. Click **OK**.
7. Assign the rule to a risk-policy and follow steps Step 7 to Step 9.

Scenario 3

You want to grant access only to employees. You want to deny access for any request from a specific region even if the user is an employee of your organization.

This scenario requires to create two separate rules: one for geolocation named `example_geolocation` and another for user profile named `example_user_profile`.

You can configure risk-based authentication for this scenario by using `Risk-based Auth Class`.

Configuration Steps:

1. Go to **Policies** > **Risk-based Policies** > **Risk Policy**.
2. Click the **Create Risk Policy** icon.
3. Under Add Risk Policy, specify the following details:

   Risk Policy Name: Specify a name.

   Policy Description: Specify the purpose of this policy.

   Assign Policy To: Select the Identity Server cluster and then configure an authentication class.
   - Select **Create Risk-based Auth Class**.
   - Specify **Class Name**.
   - Click **Save**.
4. Create a Geolocation rule and a User Profile rule.
   - **Geolocation Rule**

     Under **Policy Rules**, click **Create Rule** and specify the following values:

     NOTE: You must configure a provider in the **Geolocation** user interface for a geolocation rule to work.

     - **Rule Name**: Specify `example_geolocation`.
     - **Rule Definitions:** Select `Geolocation Rule`.

- **User Location:** Select **Is not**.

  Specify the following geolocation details of the region which you want to deny all login requests from:

  Country Code
  State Name
  State Code
  City Name
  Zip Code
  Metro Code
  Area Code
  Region Code
  Region Name

- **If rule condition is met, then**: `Allow Access and Exit Policy`
- **If rule condition is not met, add risk score:** `60`
- Click **OK**.

- **User Profile Rule**

  Under **Policy Rules**, click **Create Rule** and specify the following values:

  - **Rule Name**: Specify `example_user_profile`.
  - **Rule Definitions:** Select `User Profile`.
  - Select `employeeType`.
  - Select `Equals`.
  - Specify `Employee`.
  - **If rule condition is met, then**: `Proceed to Next Rule`
  - **If rule condition is not met, add risk score**: `60`
  - Click **OK**.

    To evaluate example_user_profile first, drag it up before example_geolocation in the rules list in the Administration Console.

5 Under Risk Levels, click **Actions** > **Add Risk Level** and create the following risk level:

For more information about risk scores, see “Risk Score” in Table 5-1, “Risk-based Authentication Terms,” on page 352.

Field	Value
Risk Score	Equals to or greater than 50
Risk Level	High
Action	Deny Access

6 Click **OK**.

7 Create an authentication method. For more information, see “Configuring a Method for an Authentication Class” on page 749.

8 Create a contract. For more information, see “Configuring a Contract for an Authentication Class” on page 749.

9 Assign the contract to the protected resource.

Scenario 4

If the user is an employee and is not located in a specific region, grant the access. If the user is an employee, but accessing from a specific region, deny the access. If the user is an employee and accessing from the specified location, but the HTTP header contains a specified email ID, grant the access.

You can configure a risk policy for this scenario by using the combination rule. A combination rule assesses more than one parameter to validate an authentication request from a user.

The rule must assess the user profile and geolocation first and consider the HTTP header condition only when the first condition evaluation fails.

You can configure risk-based authentication in this scenario by using `Risk-based Auth Class`.

Configuration Steps:

1. Go to **Policies** > **Risk-based Policies** > **Risk Policy**.
2. Click the **Create Risk Policy** icon.
3. Under Add Risk Policy, specify the following details:

   Risk Policy Name: Specify a name.

   Policy Description: Specify the purpose of this policy.

   Assign Policy To: Select the Identity Server cluster and then configure an authentication class.

   - Select **Create Risk-based Auth Class**.
   - Specify **Class Name**.
   - Click **Save**.
4. Create a Geolocation rule and a User Profile rule as a single rule.

   4a Under **Policy Rules**, click **Create Rule** and specify the following values:

   4b **Rule Name**: Specify `example_combination_rule`.

   4c Configure the geolocation rule.

   NOTE: You must configure a geolocation provider in the **Geolocation** user interface for this rule to work.

   4c1 **Rule Definitions:** Select `Geolocation Rule`.

   4c2 **User Location:** Select **Is not**.

   4c3 Specify the following geolocation details of the region which you want to deny all login requests from:

   Country Code
   State Name
   State Code
   City Name
   Zip Code
   Metro Code
   Area Code
   Region Code
   Region Name

4d Click **Combine with** to add the user profile rule.

4d1 Select `User Profile Rule`.

4d2 Under **User Attributes,** Select `employeeType` and `Equals`, and specify `Employee`.

4e Click **Combine with** to add the `HTTP Header` rule.

4e1 Select **HTTP Header Rule**.

4e2 Specify the HTTP header Name and the specific HTTP header value that you want to search for an HTTP header.

4f In **Combination Rule Definition** > **Condition Group**, click **Assign Rules** and then select user profile and geolocation rules. Select **AND** in **Group Operator**. For information about how these operators work, see “Combination Rule” in Table 5-1, “Risk-based Authentication Terms,” on page 352.

4g Click **Add Condition Group** and select `HTTP Header Rule`.

4h Select the **OR** operator for Condition Group 1 and Condition Group 2.

4i **If rule condition is met, then**: `Allow Access and Exit Policy`.

4j **If rule condition is not met, add risk score:** `50`

4k Click **OK**.

5 Under Risk Levels, click **Actions** > **Add Risk Level** and create the following risk levels:

You can define actions for a risk score or for a range of risk score. When evaluation of all conditions in a risk policy fail, the action is taken based on the accumulated risk score.

For more information, see “Risk Score” in Table 5-1, “Risk-based Authentication Terms,” on page 352.

- **Low**

Field	Value
Risk Score	Less than 30
Risk Level	Low
Action	Allow Access

- **Medium**

Field	Value
Risk Score	Between 30 and 50
Risk Level	Medium
Action	Authenticate using Trust levels

- **High**

Field	Value
Risk Score	Equals to or greater than 50
Risk Level	High
Action	Deny Access

6 Click **OK**.

7 Create an authentication method. For more information, see “Configuring a Method for an Authentication Class” on page 749.

8 Create a contract. For more information, see “Configuring a Contract for an Authentication Class” on page 749.

9 Assign the contract to the protected resource.

Scenario 5

You want to store the details of login attempts in the configured history databases and take actions based on these details in subsequent login attempts.

While configuring risk-based authentication, you can determine if you want to save the history details and the number of days for which history to consider for evaluation of the authentication attempt.

For example: Let us assume that you have enabled recording of history details and have specified that the history of last 10 days are used for evaluation before granting or denying access. If the user logs in from a different geolocation, additional authentication is requested as the risk is high. The risk evaluation details are stored in the database. Next time the user logs in from the same geolocation, the historical details for the last ten days are checked to see if there are details about a login attempt from the same geolocation. As the geolocation details exist in the database, the user is granted access without being prompted for additional authentication.

You can enable recording of user history only for a risk policy that uses `Risk-based Auth Class`.

For more information about how to enable recording of user history, see “Configuring User History” on page 754.

Understanding Risk Score Calculation

A risk score is assigned when a rule is added to a risk policy. This risk score indicates the priority and criticality of the rule.

For example, if you have configured a set of rules, but you want one rule to be the most important rule, assign it a higher risk score compared to the other rules. If the rule evaluation is successful, the risk score is set as zero.

If a rule evaluation is not successful, the risk score is set as the value of the rule. If you have configured multiple rules, the total risk score is the sum of risk scores of all the failed rules.

- “Scenario 1” on page 360
- “Scenario 2” on page 361

Scenario 1

Let us assume that you have created two rules to validate login requests to a financial application. You have determined that Rule 1 is the most critical rule and want users to gain access when this rule is evaluated.

Table 5-2 Risk Rules

Rules	Risk Score	If rule condition is met, then
Rule 1	50	Allow access and exit policy
Rule 2	30	Return risk level low

Depending on the risk score returned after evaluation of rule, risk level is assigned and action is taken.

Table 5-3 Risk Scores and Risk Levels

Total Risk Score	Risk Level	Action
31-80	Medium	Additional authentication must be requested.
0-30	Low	Allow access.

The following table describes how the rules are evaluated:

Table 5-4 Risk Score Calculation for the Rules

Scenario	Details	Total Risk Score	Action
Rule 1 is successfully evaluated.	Rule 2 is not considered for rule processing as Rule 1 is configured to exit the policy when condition is met.	0	Access is allowed.
Rule 1 and Rule 2 fail.	In this case, the total risk score is 80 as both the rules have failed.	80	Additional authentication is requested.

Scenario 2

You have created three rules to access login requests to a financial application. All the rules must evaluate successfully to grant access to the user.

Table 5-5 Risk Rules

Rules	Risk Score
Rule 1	50
Rule 2	30
Rule 3	10

Depending on the risk score returned after evaluation of rule, risk level is assigned and action is taken.

Table 5-6 Risk Scores and Risk Levels

Total Risk Score	Risk Level	Action
0-30	Low	Allow access
31-50	Medium	Additional authentication
51-100	High	Deny access

The following table describes how the rules are evaluated:

Table 5-7 Risk Score Calculation for the Rules

Scenario	Details	Total Risk Score	Action
Rule 1, Rule 2, and Rule 3 are successfully evaluated.	As all the rules are evaluated without errors, the risk score is 0.	0	Access is allowed.
Rule 1 evaluates successfully, but Rule 2 and Rule 3 fail.	The risk score is the value assigned to the rule that failed. In this case, the risk score is 40.	40	Additional authentication is requested.
Rule 1 fails, but Rule 2 and Rule 3 evaluate successfully.	The risk score is the value assigned to the rule that failed. In this case, the risk score is 50.	50	Additional authentication is requested.
Rule 2 evaluates successfully, but rule 1 and rule 3 fail.	The risk score is the sum of risk scores of all failed rules. In this case, the risk score is 60.	60	Access is denied.
Rule 2 fails, but rule 1 and rule 3 evaluate successfully.	The risk score is the sum of risk scores of all failed rules. In this case, the risk score is 30.	30	Access is allowed.
All the rules fail.	The risk score is the sum of risk scores of all failed rules. In this case, the risk score is 90.	90	Access is denied.

Configuring Risk-based Authentication

For information about configuring risk-based authentication, see “Risk-Based Policies” on page 745.

Configuring an External Database to Store User History

- “Configuring MySQL Database” on page 363
- “Configuring Oracle Database” on page 363
- “Configuring Microsoft SQL Server” on page 364

Configuring MySQL Database

1. Unzip the `/opt/novell/nids/lib/webapp/WEB-INF/RiskDBScripts.zip` file containing script to extend the database and sample configuration files. The file is located at the following locations:
2. On the MySQL server, execute the following command to create database objects for risk-based authentication:

   `mysql -h host -u username -p password netiq_risk_mssql_install.sql`
3. On the Identity Server, update the `/opt/novell/nam/idp/webapps/nidp/WEB-INF/classes/hibermate.cfg.xml` file with server IP address and user credentials. The file is located at the following locations:

   Refer to the sample configuration files packaged with the `RiskDBScript.tar` file for details on changing the `connection` property.
4. Download the JDBC connector for the MySQL database from MySQL.com.
5. Copy the JDBC connector to the `/opt/novell/nids/lib/webapp/WEB-INF/lib/` folder.
6. Restart the Identity Server.

Configuring Oracle Database

1. Unzip the `/opt/novell/nids/lib/webapp/WEB-INF/RiskDBScripts.zip` file containing script to extend the database and sample configuration files. The file is located at the following locations:
2. On the Oracle server, execute the following script to create database objects for risk-based authentication:

   `netiq_risk_oracle_install.sql`
3. On the Identity Server, update the `/opt/novell/nam/idp/webapps/nidp/WEB-INF/classes/hibermate.cfg.xml` file with server IP address and user credentials. The file is located at the following locations:

   Refer to the sample configuration files packaged with the `RiskDBScript.tar` file for details on changing the `connection` property.
4. Download the JDBC connector for the Oracle database from Oracle.com.
5. Copy the JDBC connector jar to the `/opt/novell/nids/lib/webapp/WEB-INF/lib/` folder.
6. Restart the Identity Server.

Configuring Microsoft SQL Server

1. Unzip the `/opt/novell/nids/lib/webapp/WEB-INF/RiskDBScripts.zip` file containing script to extend the database and sample configuration files. The file is located at the following locations:
2. On the SQL Server, execute the following script to create database objects for risk-based authentication:

   `netiq_risk_sql_server_install.sql`
3. On the Identity Server, update the `/opt/novell/nam/idp/webapps/nidp/WEB-INF/classes/hibermate.cfg.xml` file with server IP address and user credentials. The file is located at the following locations:

   Refer to the sample configuration files packaged with the `RiskDBScript.tar` file for details on changing the `connection` property.

4 Download the JDBC connector for the SQL Server database from Microsoft.com (https://www.microsoft.com/en-in/download/details.aspx?id=11774).

5 Copy the JDBC connector file `sqljdbc42.jar` to the `/opt/novell/nids/lib/webapp/WEB-INF/lib/` folder.

6 Restart the Identity Server.

Troubleshooting Risk Rule Configuration

The following sections describe how to troubleshoot rule configuration:

- “Understanding How to Use the Validate Tool to Emulate Total Risk Score and Risk Levels” on page 364
- “Understanding How To Use the Risk Rule Validation Utility To Troubleshoot Rule Configuration” on page 366
- “Troubleshooting Rule Evaluation Details By Using the Log File” on page 366

Understanding How to Use the Validate Tool to Emulate Total Risk Score and Risk Levels

After configuring a risk policy and the corresponding risk scores and actions, use **Validate** to emulate total risk score, risk level, and action in event of rule failure. Based on the results, you can modify the configuration, if required.

Let us consider a case where you have configured a risk policy that includes five rules. The rules and the corresponding risk scores are as follows:

Table 5-8 *Sample Risk Policy Configuration: Rules*

Risk Policy Name	Rule	Risk Score
Demo_RiskPolicy	Demo_InNetworkAtOfficeHours	20
	Demo_InternalUser	20
	Demo_KnownDevice	20
	Demo_PayrollSiteCookie	20
	Demo_UserProfile	20

Table 5-9 *Sample Risk Policy Configuration: Risk Scores and Risk Levels*

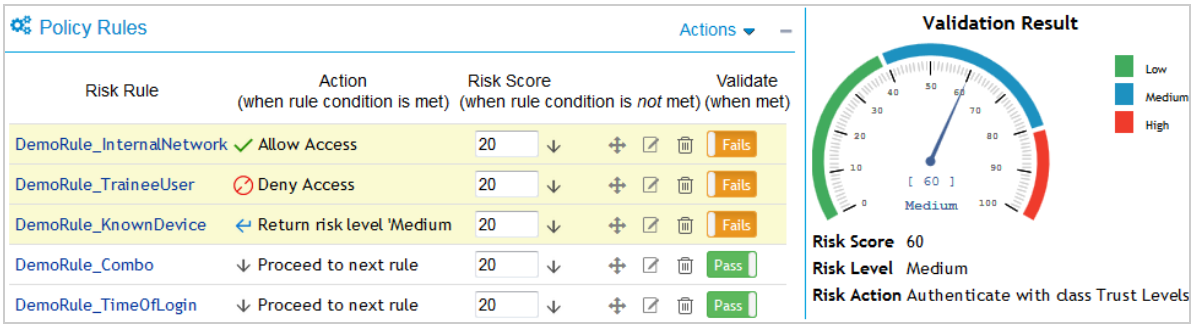
Risk Score	Risk Level	Action
Less than 30	Low	Allow access
Between 30 to 60	Medium	Authenticate with class Trust Levels
Greater than 60	High	Deny access

Now, open the risk policy for which you want to emulate total risk score, risk level, and action in event of rule failure. In the risk policy page, click **Actions** > **Toggle Validate**. Specify the rules as pass or fail to see the result along with a graphical reperesentation.

For example, specify pass and fail for rules as follows:

Rule	Condition
Demo_InNetworkAtOfficeHours	Failed
Demo_InternalUser	Failed
Demo_KnownDevice	Failed
Demo_PayrollSiteCookie	Passed
Demo_UserProfile	Passed

In this case, the validation result is as follows:



You can similarly specify any other rule as failed or passed to emulate the risk score and risk levels.

Understanding How To Use the Risk Rule Validation Utility To Troubleshoot Rule Configuration

After configuring a risk policy, you can use the Risk Rule Validation utility to evaluate the configuration of rules. This helps you understand how rules are evaluated in a risk policy.

During rule evaluation if there is a match with the values configured for the rules, the rule evaluation is successful. If no match is found, the rule evaluation fails.

Using the Risk Rule Validation Utility to Test Risk Configuration

To use the risk rule validation utility for testing risk configuration, perform the following steps:

1. In the browser address bar, type the following URL:

   `https://<identity-server-base-url>:port/nidp/test/risk`

   For example: `https://10.1.1.1:8443/nidp/test/risk`
2. Specify the credentials to log in.
3. Select a risk policy for evaluation. Click **Submit**. The risk score, risk category evaluation results and HTTP request header and related information are displayed.
4. [Optional] If you have logged in with administrator privileges, click **Details** to view details about risk configuration.

   NOTE: The Risk Rule Validation utility does not display details if **Record User History** is enabled and a user profile rule is configured.

Troubleshooting Rule Evaluation Details By Using the Log File

You can troubleshoot rule evaluation details by performing the following details:

- “Prerequisite” on page 366
- “Using Logs to Understand How Rules are Evaluated” on page 366

Prerequisite

Ensure that you have enabled logging at the application level. For more information see, “Enabling Logging for Risk-Based Authentication” on page 756.

Using Logs to Understand How Rules are Evaluated

If you encounter any error during risk-based authentication, check the log files to review the error code. The log file location is:

Linux: `/opt/novell/nam/idp/logs/catalina.out`

Windows: `\Program Files (x86)\Novell\Tomcat\logs\stdout.log`

Consider a scenario where you have three rules configured as described in Table 5-8. Using this scenario as an example let us see how we can use the details in `catalina.out` file to understand how rules are evaluated.

- “Scenario 1: User Profile Rule Fails” on page 367
- “Scenario 2: User Profile Rule Evaluates Successfully” on page 367
- “Scenario 3: Two rules fail and the user is asked to authenticate using additional authentication” on page 368
- “Scenario 4: All the Rules Fail” on page 369

Scenario 1: User Profile Rule Fails

In this scenario the User Profile rule fails to evaluate successfully. All other rules in the risk policy evaluate successfully.

The following tracelist detail from `catalina.out` file provides more information about rule evaluation, risk score, and action:

Figure 5-5 *Tracelist providing information about rule evaluation*

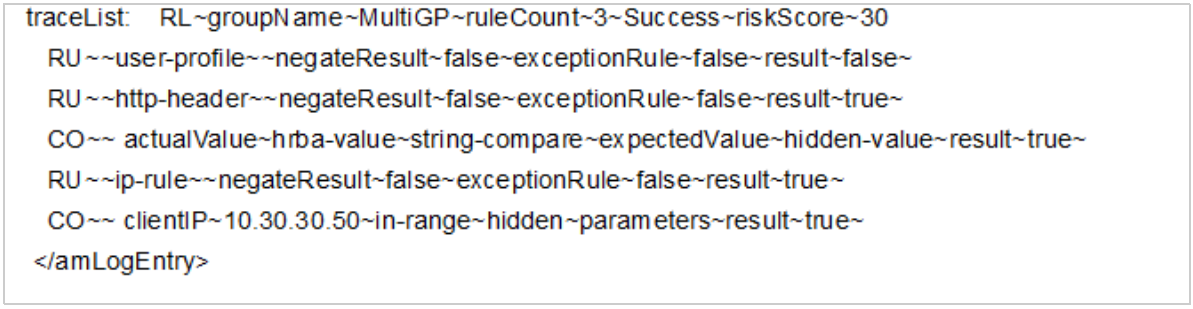
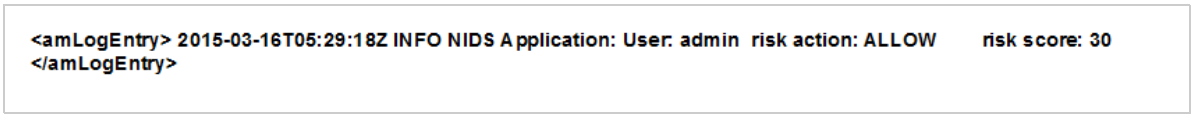


Table 5-10 *Description of details recorded in the catalina.out file.*

Entry	Description
user-profile~result~false	Indicates that user profile rule failed and the risk score of 30 is added to the total risk score.
http-header~result~true	Indicates that the HTTP header rule evaluated successfully.
ip-rule~result~true	Indicates that the IP address rule evaluated successfully.

Figure 5-6 *Tracelist providing information about risk level and action*



This log entry indicates that the as per the risk level/action configuration, the action taken is to allow authentication to the user and the risk score is 30.

Scenario 2: User Profile Rule Evaluates Successfully

In this scenario the User Profile rule evaluates successfully. As this rule is a configured to exit when conditions is met, all other rules in the risk policy are not considered for evaluation.

The following tracelist detail from `catalina.out` file provides more information on the rule evaluation, risk score and action.

Figure 5-7 *Tracelist providing information about rule evaluation*

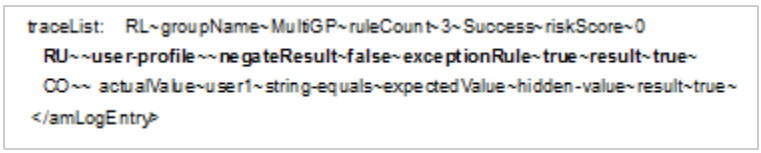


Table 5-11 *Description of details recorded in catalina.out file*

Entry	Description
`user-profile~result~true`	Indicates that user profile rule evaluated successfully.

Figure 5-8 *Tracelist providing information about risk level and action*



This log entry indicates that the as per the risk level/action configuration, the action taken is to allow authentication to the user and the risk score is 0.

Scenario 3: Two rules fail and the user is asked to authenticate using additional authentication

In this scenario User Profile rule and the IP address rule fail to evaluate successfully. The HTTP header rule evaluates successfully.

The following tracelist detail from `catalina.out` file provides more information on the rule evaluation, risk score and action.

Figure 5-9 *Tracelist providing information about rule evaluation*

```
traceList:   RL~groupName~MultiGP~ruleCount~3~Success~riskScore~55
  RU~~user-profile~~negateResult~false~exceptionRule~false~result~false~
  RU~~http-header~~negateResult~false~exceptionRule~false~result~true~
  CO~~ actualValue~hrba-value~string-compare~expectedValue~hidden-value~result~true~
  RU~~ip-rule~~negateResult~false~exceptionRule~false~result~false~
  CO~~ clientIP~                    ~in-range~hidden~parameters~result~false~
 </amLogEntry>
```

Table 5-12 *Description of details recorded in the catalina.out file*

Entry	Description
user-profile~result~false	Indicates that user profile rule failed and the risk score of 30 is added to the total risk score.
http-header~result~true	Indicates that the HTTP header rule evaluated successfully.
ip-rule~result~false	Indicates that the IP address rule failed and the risk score of 25 is added to the total risk score.

Figure 5-10 *Tracelist providing information about risk level and action*

```
<amLogEntry> 2015-03-16T05:28:56Z INFO NIDS Application: User: admin  risk action:  STEP_UP/Additional authentication      risk score: 55
</amLogEntry>
```

This log entry indicates that the as per the risk level/action configuration, the action taken is additional authentication and the risk score is 55.

Scenario 4: All the Rules Fail

In this scenario all the rules fail to evaluate successfully.

The following tracelist detail from `catalina.out` file provides more information on the rule evaluation, risk score and action.

Figure 5-11 *Tracelist providing information about rule evaluation*

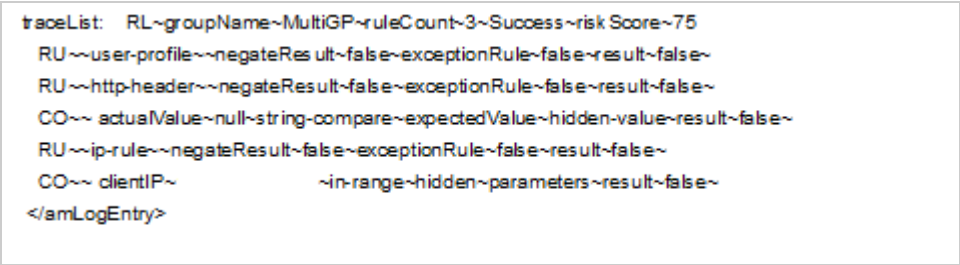


Table 5-13 *Description of details recorded in the catalina.out file*

Entry	Description
user-profile~result~false	Indicates that user profile rule failed and the risk score of 30 is added to the total risk score.
http-header~result~false	Indicates that the HTTP header rule failed and the risk score of 20 is added to the total risk score.
ip-rule~result~false	Indicates that the IP address rule failed and the risk score of 25 is added to the total risk score.

Figure 5-12 *Tracelist providing information about risk level and action*

This log entry indicates that as per the risk level/action configuration, the action is to deny access to the user and the risk score is 75.

Managing Direct Access to the Identity Server

Users usually log into the Identity Server when they request access to a Web resource. The Access Gateway redirects users from the resource to the Identity Server to provide the required credentials for the resource. After they are authenticated, they are not prompted for credentials again, unless a resource requires credentials that they haven't already supplied.

Users can log directly into the Identity Server and access either the default user portal or the legacy user portal. Alternatively, they can access information about the available Web Services Description Language (WSDL) services.

Logging in to the Default User Portal

Access Manager Appliance 4.2 default user portal page is the user portal page that contains appmarks. This is a different portal page than for prior release. For more information, see Access Manager Appliance 4.2 Release Notes.

User log directly in to the Identity Server through the default user portal page when they access the following URL:

`https://`*`NAM-Base-URL`*`:8443/nidp/portal`

IMPORTANT: The Access Manager domain name for the IDP must be resolvable through DNS or the `hosts` file on the IDP.

You can customize the default user portal page through Branding in the Administration Console. For more information, see “Changing the Branding of the User Portal Page” on page 769.

Logging in with the Legacy Customized Portal

If you have customized the legacy user portal page, and want to keep the customized pages, you must perform some manual steps. For more information, see Maintaining Customized JSP Files for Identity Server.

Managing Authentication Cards

The default user portal page prompts the user to authentication with the credentials required for the default contract. When users log directly into the Identity Server, the users must use the default card for authentications. The menu in the top left corner displays all available cards, when the users click the menu.

On a newly installed system, the menu displays cards for all the authentication contracts that are installed with the system. To avoid confusing your users, you need to disable the **Show Card** option for the contracts you do not want your users to use. Also, ensure that you modify the default contract to match a card that the Administration Console displays.

If you display multiple cards, users can use different credentials to authenticate multiple times by selecting another authentication card and entering the required credentials. This is only useful if the credentials grant the user different roles or authorize access to different resources.

If you have configured the Identity Server to be a service provider and have established a trusted relationship with one or more identity providers, the cards of these trusted identity providers appear in the menu under **REMOTE LOGINS**, Your users can use the identity provider’s authentication card to federate their account at the identity provider with their account at the service provider. When they federate an account, they are telling the service provider to trust the authentication established at the identity provider. This enables single sign-on between the providers. The card can also be used to defederate the accounts. On the authentication card, click **Defederate**.

If you have configured the Identity Server to be an identity provider for service providers, in the menu in the upper right corner of the user portal page contains a **Federations** options that displays a Federations page. From this page, users can federate and deferated their accounts with trusted service providers.

To edit the default contracts:

1. Log in to the Administration Console.
2. Click **Devices** > **Identity Servers** > **Edit** > **Local** > **Contracts** > *Name of Contract* > **Authentication Card**.
3. Unselect the **Show Card** option, then click **OK** to save the change.
4. On the **Local** page for the Identity Service, click **Defaults**.
5. Modify the contract to match the cards that the Administration Console displays.
6. Click **OK** to save the changes.

Specifying a Target

You can specify a target for the new and legacy user portal pages. You must specify a target for the following conditions:

- You want to direct the users to a specific URL after the users log in to the Identity Server.
- You do not want users to have access to the user portal page.

Use one of the following methods to specify the target:

- **Specify a Target in the URL:** You can have your users access the Identity Server with a URL that contains the desired target. For example:
  - **Default User Portal:** `https://domain.com:8443/nidp/portal?target=http://www.acme.com`
  - **Legacy User Portal:** `https://domain.com:8443/nidp/app?target=http://www.acme.com`

  where *domain.com* is the DNS name of your Identity Server. In this example, the users would see the Acme Web site after logging in.
- **Specify a Hidden Target on your Form:** If you have your own login form to collect credentials and are posting these credentials to the Identity Server, you can add a hidden target to your login form. When authentication succeeds, Access Manager Appliance directs the user to this target URL. This entry on your form should look similar to the following:

  ```
  <input type="hidden" target="http://www.acme.com">
  ```

  These methods work only when the user's request is for the user portal (`/nidp/portal` or `/nidp/app`). If the user's request is a redirected authentication request for a protected resource, the protected resource is the target and cannot be changed.

Blocking Access to the User Portal Page

This information is for the legacy user portal page only. You cannot block the default user portal page.

If a user is already authenticated and accesses the Identity Server, the user is presented with the Identity Server user portal page.

This page provides information about the logged-in user:

- Any federations this user has established with third-party service providers
- Identity attributes such as Liberty Personal or employee profile attributes, Access Manager credential, or custom profile attributes
- Policy attributes that users or administrators have selected to share with other service providers

You might want to prevent users from seeing this page for the following reasons:

- **Security:** Users accessing this page have access to sensitive information that administrators might want to restrict such as the user's attributes and federations with other third-party SAML or Liberty providers.
- **Help Desk Support:** Most users have no need to access the information presented in this page. As a result, they might be confused if they see it. By preventing access to the page, any potential calls into the help desk are avoided.

When you enable the legacy, mode. all end user login pages use the legacy UI. However, there are several places where issues the URL`/nidp` and shows the portal page. This is because in the `index.html` file for the nidp webapp there is a redirect from `/nidp` to `/nidp/portal`.

To block the legacy user portal page:

1. Ensure that you create the `WEB-INF/legacy` directory. For more information, see “Logging in with the Legacy Customized Portal” on page 370.
2. As a system administrator, edit the `/webapp/index.html` file.
3. Change the line:

   ```
   <meta http-equiv="refresh" content="0; URL=portal">
   ```

   to:

   ```
   <meta http-equiv="refresh" content="0; URL=app">
   ```

4. Save the `/webapp/index.html` file.

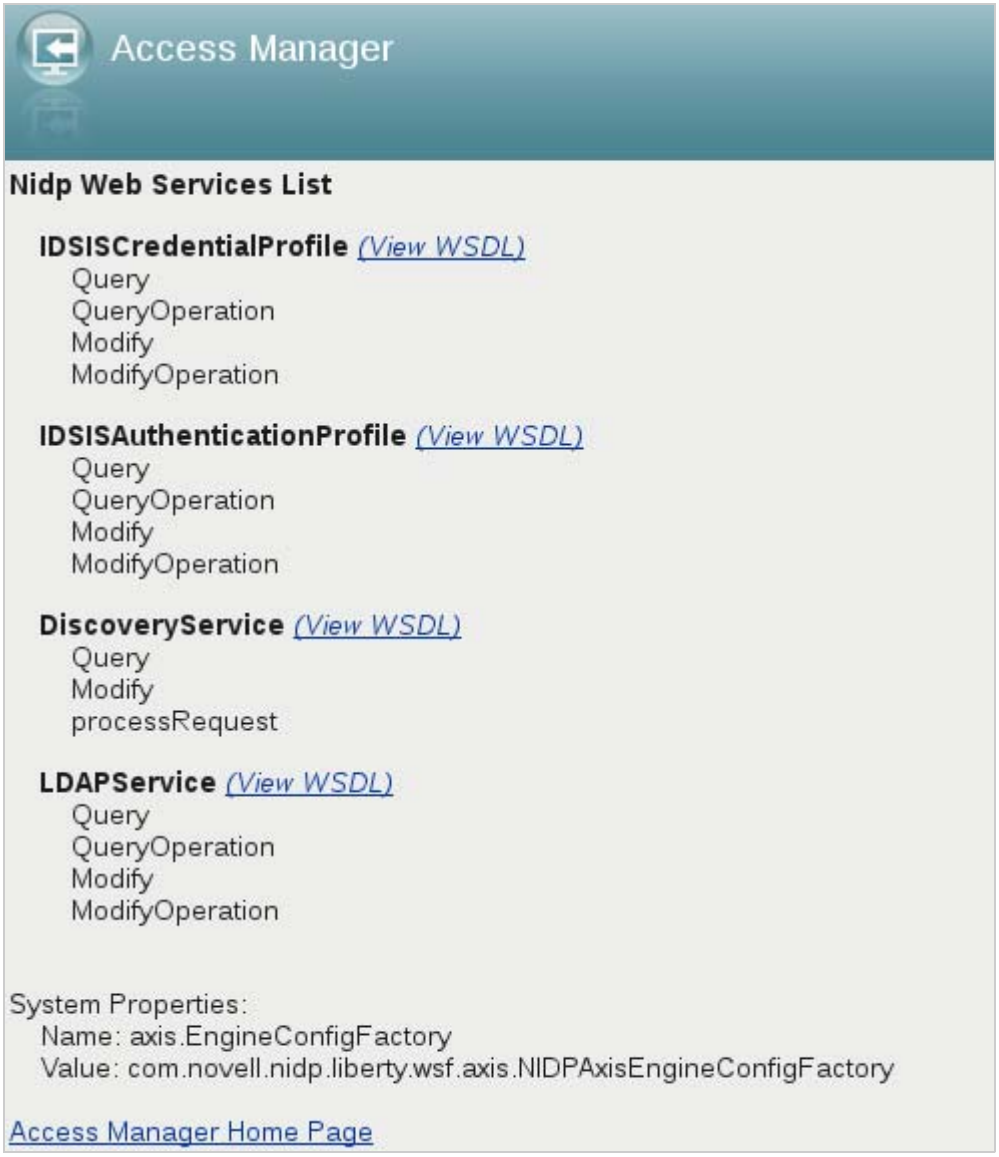
After this change takes place, the only way to access the legacy user portal page is to enter the URL in the browser of `/nidp/portal`.

Blocking Access to the WSDL Services Page

Users can access the WSDL services page when they enter the base URL of the Identity Server in their browsers with the path to the Services page. For example, if your base URL is `http://bfrei.provo.novell.com:8080/nidp`, the users can access the services page by using `http://bfrei.provo.novell.com:8080/nidp/services`.

The Services page contains the following information and links:

Figure 5-13 *WSDL Services Page*



The amount of information displayed on this page depends upon the profiles you have enabled. To enable profiles, click **Devices > Identity Servers > Edit > Liberty > Web Service Provider**.

If you do not want your users to have access to this page, you can block access.

1. In the Administration Console, click **Devices > Identity Servers > Edit** > **Options**.
2. Click **New**. Specify the following details:

Property Type	Property value
Select WSF SERVICES LIST.	Select full to enable users to access the Services page. Select 404 to return an HTTP 404 status code: Not Found Select 403 to return an HTTP 403 status code: Forbidden Select empty to return an empty services list

3 Restart Tomcat for your modifications to take effect:

```
/etc/init.d/novell-idp restart Or
rcnovell-idp restart
```

Configuring Federated Authentication

Federation allows a user to associate two accounts with each other. This allows the user to log into one account and access the resources of the other account without logging in to the second account. It is one method for providing single sign-on when a user has accounts in multiple user stores.

- “Configuring Federation” on page 375
- “Service Provider Brokering” on page 394
- “Configuring User Identification Methods for Federation” on page 416
- “Configuring SAML 2.0” on page 423
- “Configuring SAML 1.1” on page 458
- “Configuring Liberty” on page 461
- “Configuring Liberty Web Services” on page 468
- “Configuring WS Federation” on page 486
- “Configuring WS-Trust Security Token Service” on page 511
- “Configuring OAuth and OpenID Connect” on page 532
- “Configuring Authentication Through Federation for Specific Providers” on page 568
- “Integrating Amazon Web Services with Access Manager” on page 572
- “Configuring Single Sign-On for Office 365 Services” on page 575

Configuring Federation

This section describes what is federation, how to configure federation, and how to set up federation with third-party providers. Topics include:

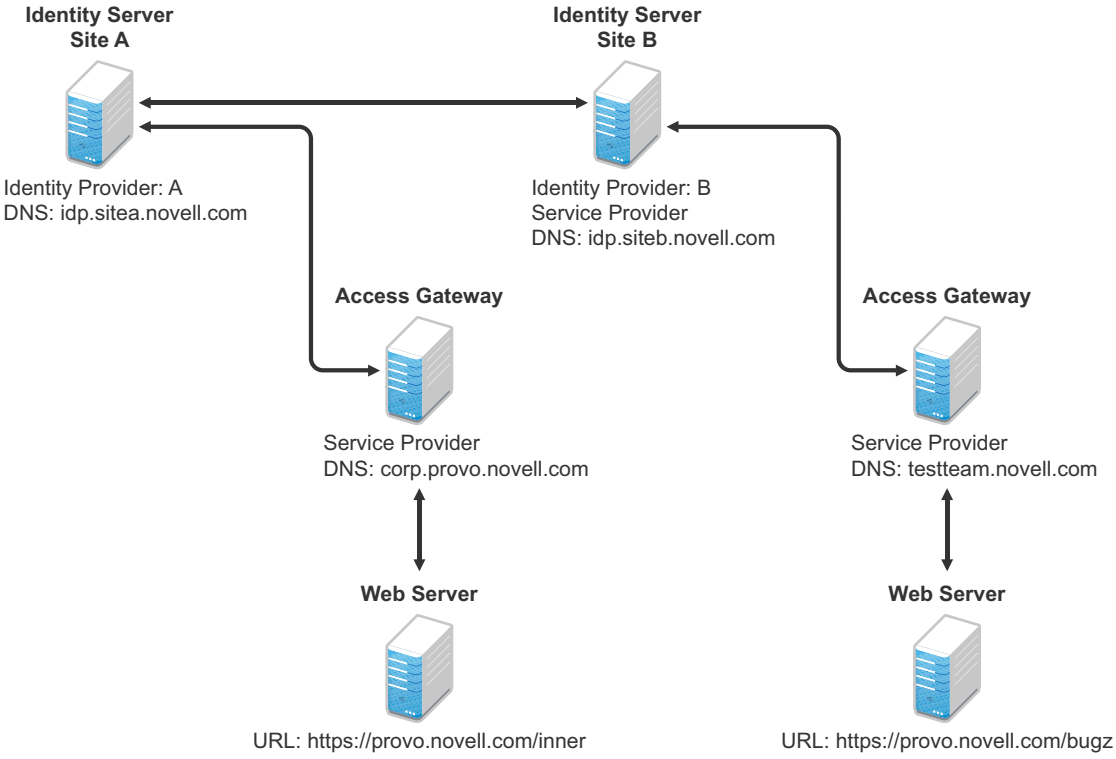
- “Understanding a Simple Federation Scenario” on page 375
- “Configuring Federation” on page 376
- “Sharing Roles” on page 387
- “Setting Up Federation with Third-Party Providers” on page 393

Understanding a Simple Federation Scenario

Suppose Company A has a centralized user store that does the authentication for most of the company's internal resources on its inner Web site. But Company A also has a customer feedback application that employees and customers need access to, and for this application, a second user store has been created. This user store contains both employee and customer user accounts. The centralized user store cant be used, because it can contain only employee accounts. This means that the employee must log in to both accounts to access both the inner Web site and the customer feedback application. With federation, the employee can access the resources of both sites by using a single login.

Figure 5-14 illustrates such a network configuration where the user accounts of Site A are configured to federate with the user accounts at Site B.

Figure 5-14 *Using Federated Identities*



In this configuration, Site A is the Identity Server for the corporate resources, and the employees authenticate to this site and have access to the resources on the Web server with the URL of `https://provo.novell.com/inner`. Site B is the Identity Server for the Bugzilla application, and both employees and customers authenticate to this site to have access to the resources of the Web server with the URL of `https://provo.novell.com/bugz`. After an account has been federated, the user can log in to Site A and have access to the resources on the Web servers of both Site A and Site B.

In this scenario, Site B is not as secure a site as Site A, so federation is configured to go only one way, from Site A to Site B. This means that users who log in to Site A have access to the resources at Site A and B, but users who log in to Site B have access only to the resources at Site B. Federation can be configured to go both ways, so that it does not'matter whether the user logs into Site A or Site B. When federation is configured to be bidirectional, both sites need to be equally secure.

The Access Gateways in Figure 5-14 are service providers and are configured to use the Identity Servers as identity providers. The trusted relationship is automatically set up for you when you specify authentication settings for the Access Gateway and select an Identity Server Cluster.

Federation can be set up between providers in the same company or between providers of separate companies. For example, most companies have contracts with other companies for their user's health benefits and retirement accounts. Their users have accounts with these companies. These accounts can be federated with the user's employee account when both companies agree to set up the trusted relationship.

Configuring Federation

Federation requires the configuration of a trusted relationship between an identity provider and a service provider. Figure 5-15 illustrates setting up federation between two identity servers, because a NetIQ Identity Server can act as either an identity provider or a service provider.

Figure 5-15 *Configuring Trust Between Site A and Site B*



Site A must be configured to trust Site B as a service provider, and Site B must be configured to trust Site A as an identity provider. Until this two-way trust is established, federation cannot occur.

Before setting up a trusted relationship, you must make the following decisions:

Protocol: The Identity Server supports SAML 1.1, SAML 2.0, and Liberty. You need to decide which of these protocols to use. If no user interaction is needed, SAML 1.1 is probably a good choice. The SAML 2.0 and Liberty protocols permit user interaction when federating. The user decides whether to federate (link) the accounts and must be logged in at both sites to accomplish this. Liberty offers an additional service, not available with SAML 2.0, that allows the user to select attributes that can be shared with the service provider.

The instructions in this documentation, starting in "Prerequisites" on page 377, use the Liberty protocol. They also indicate how to configure for the SAML 2.0 and SAML 1.1 protocols.

Trust Relationship: You need to decide whether the trusted relationship is going to be from Site A to Site B, from Site B to Site A, or bidirectionally from Site A to Site B and from Site B to Site A. Federation is set up to go from the most secure site to the less secure site. The only time federation is set up to be bidirectional is when both sites are equally secure. The scenario described in Figure 5-14 on page 375 is an example of a trusted relationship that you would want to go only one way, from Site A to Site B, because Site B is not as secure as Site A.

The instructions, starting in "Prerequisites" on page 377, explain how to set up the trusted relationship between Site A and Site B. You can easily modify them to set up the bidirectional trust relationships by substituting Site B for Site A (and vice versa) in the instructions and then repeating them for Site B

Attributes to Share: You need to decide whether there are user attributes or roles at Site A that you want to share with Site B. The attributes from Site A can be used to identify the users at Site B. Other attributes might be needed to access protected resources, for example, to satisfy the requirements of an Identity Injection policy.

For all the protocols, “Sharing Roles” on page 387 explains how to share the roles at Site A with Site B. For the SAML 1.1 protocol, the instructions starting in “Prerequisites” on page 377 use the LDAP mail attribute to share the user’s e-mail address.

User Identification: You need to decide how assertions can be used to map users from Site A to users at Site B. The Identity Server supports four methods:

- **Temporary:** This method allows the user access to Site B solely from the credentials of Site A. No effort is made to map the user to a user account at Site B. A temporary account is set up for the user on Site B, and when the user logs out, the account is destroyed.
- **Login:** This method requires that the user have login credentials at both Site A and Site B, and when logged in at both sites, the user can select to federate the accounts.
- **Mapped Attributes:** This method requires that the sites share attributes and that these attributes are used to create a matching expression that determines whether the user accounts match. For an added security check, the first time the accounts are matched, the user is asked to verify the match by supplying the password for Site B.

  If the match fails, you can allow the federation to fail or you can configure the method to allow the user to use the Login method or the Provisioning method.
- **Provisioning:** This method allows the user to create a new, permanent account at Site B.

The configuration instructions, starting in “Prerequisites” on page 377, use the Login method for the SAML 2.0 and Liberty protocols and Mapped Attributes method for the SAML 1.1 protocol.

The instruction for setting up a trusted relationship between two NetIQ Identity Servers have been divided as follows:

- “Prerequisites” on page 377
- “Establishing Trust between Providers” on page 378
- “Configuring SAML 1.1 for Account Federation” on page 384

Prerequisites

- ❒ A basic Access Manager Appliance configuration with the Identity Server and Access Gateway configured for SSL.

  This can be the one you set up using the instructions in Chapter 3, “Setting Up a Basic Access Manager Appliance Configuration,” on page 49. For SSL configuration, see Chapter , “Enabling SSL Communication,” on page 819.

  The Identity Server from this configuration becomes Site B in Figure 5-15.
- ❒ A second Identity Server with a basic configuration, an LDAP user store, and SSL. This Identity Server is configured to be Site A in Figure 5-15.
- ❒ Time synchronization must be set up for all the machines, or authentication can fail if assertions expire before they can be used.
- ❒ A DNS server must be configured to resolve the DNS names of Site A, Site B, and the Access Gateways.
- ❒ (Recommended) Logging has been enabled on the Identity Servers of Site A and Site B. See “Configuring Logging for Identity Server” on page 852. Make sure that you enable at least application and protocol (Liberty, SAML 1, or SAML 2.0) logging at an Info level or higher.

Establishing Trust between Providers

To set up this very basic example of federation, complete the following tasks.

- “Configuring Site A to Trust Site B as a Service Provider” on page 378
- “Configuring Site B to Trust Site A as an Identity Provider” on page 379
- “Verifying the Trust Relationship” on page 381
- “Configuring User Authentication” on page 382

Configuring Site A to Trust Site B as a Service Provider

To establish trust between Site A and Site B, you must perform two tasks:

- The providers must trust the certificates of each other so you need to import the trusted root certificate of Site B to Site A.
- You must also import the metadata of Site B to Site A. The metadata allows Site A to verify that Site B is truly Site B when Site B sends a request to Site A.

The following instructions explain how to import the certificate and the metadata:

1. Log in to the Administration Console for Site A.

   The configuration for Site A can be created in the same Administration Console as Site B; it cannot be configured to be a cluster member of Site B.

2. Import the trusted root certificate of Site B into the NIDP trust store of Site A:

   2a Click **Devices** > **Identity Servers** > **Edit** > **Security** > **NIDP Trust Store**.

   2b In the Trusted Roots section, click **Auto-Import From Server**, then fill the following fields:

   Server IP/DNS: Specify the IP address or DNS name of Site B. For Site B in Figure 5-15 specify the following:

   ```
   idp.siteb.novell.com
   ```

   Server Port: Specify 8443.

   2c Click **OK**, then specify an alias for the certificate (for example, SiteB).

   You will get two certificate options: Root CA Certificate andServer certificate. We recommend you to select Root CA Certificate.

   2d Examine the trusted root that is selected for you.

   If the trusted root is part of a chain, make sure you select the parent and all intermediate trusted roots.

   2e Click **OK**.

   The trusted root certificate of Site B is added to the NIDP trust store.

   2f Click **Close**.

   2g Click **Devices** > **Identity Servers**, then update the Identity Server.

   Wait for the health status to return to green.

3. Configure a service provider for Site A:

   3a Click **Identity Servers** > **Edit** > **Liberty** [or **SAML 2.0** or **SAML 1.1**].

   3b Click **New**, select **Service Provider**, then fill the following fields:

   Name: Specify a name for the provider. If you plan on configuring more than one protocol, include the protocol as part of the name, such as, SiteB_Liberty

Metadata URL: Specify the URL of the Liberty metadata on Site B. For Site B in Figure 5-15, specify the following:

```
http://idp.siteb.novell.com:8080/nidp/idff/metadata
```

This example uses port 8080 to avoid any potential certificate problems that occur when the Identity Server and the Administration Console are installed on separate machines.

SAML 2.0: If you are using SAML 2.0, the metadata path is `/nidp/saml2/metadata`. For Site B in Figure 5-15, specify the following for SAML 2.0:

```
http://idp.siteb.novell.com:8080/nidp/saml2/metadata
```

SAML 1.1: If you are using SAML 1.1, the metadata path is `/nidp/saml/metadata`. For Site B in Figure 5-15, specify the following for SAML 1.1:

```
http://idp.siteb.novell.com:8080/nidp/saml/metadata
```

3c Click **Next** > **Finish** > **OK**.

3d Update the Identity Server.

Wait for the health status to return to green.

4 Continue with "Configuring Site B to Trust Site A as an Identity Provider" on page 379.

Configuring Site B to Trust Site A as an Identity Provider

The following instructions explain how to import the trusted root certificate and metadata of Site A into the configuration for Site B.

1 Log in to the Administration Console for Site B.

The configuration of Site B can be created in the same Administration Console as Site A; it cannot be configured to be a cluster member of Site A.

2 Import the trusted root certificate of Site A into the NIDP trust store of Site B.

2a Click **Devices** > **Identity Servers** > **Edit** > **Security** > **NIDP Trust Store**.

2b In the Trusted Roots section, click **Auto-Import From Server**, then fill the following fields:

Server IP/DNS: Specify the IP address or DNS name of Site A. For Site A in Figure 5-15, specify the following:

```
idp.sitea.novell.com
```

Server Port: Specify 8443.

2c Click **OK**, then specify an alias for the certificate (for example, SiteA).

You will get two certificate options: Root CA Certificate andServer certificate. We recommend you to select Root CA Certificate.

2d Examine the trusted root that is selected for you.

If the trusted root is part of a chain, make sure you select the parent and all intermediate trusted roots.

2e Click **OK**.

The trusted root certificate of Site A is added to the NIDP trust store.

2f Click **Close**.

2g Click **Identity Servers** > **Update** > **OK**.

Wait for the health status to return to green.

3 Configure an identity provider for Site B.

3a Click **Identity Servers** > **Edit** > **Liberty** [or **SAML 2.0** or **SAML 1.1**].

3b Click **New**, select **Identity Provider**, then fill the following fields:

Name: Specify a name for the provider. If you plan on configuring more than one protocol, include the protocol as part of the name, such as SiteA_Liberty

Metadata URL: Specify the URL of the Liberty metadata on Site A. For Site A in Figure 5-15, specify the following:

```
http://idp.sitea.novell.com:8080/nidp/idff/metadata
```

This example uses port 8080 to avoid any potential certificate problems that occur when the Identity Server and the Administration Console are installed on separate machines.

SAML 2.0: If you are using SAML 2.0, the metadata path is `/nidp/saml2/metadata`. For Site A in Figure 5-15, specify the following for SAML 2.0:

```
http://idp.sitea.novell.com:8080/nidp/saml2/metadata
```

SAML 1.1: If you are using SAML 1.1, the metadata path is `/nidp/saml/metadata`. For Site B in Figure 5-15, specify the following for SAML 1.1:

```
http://idp.siteb.novell.com:8080/nidp/saml/metadata
```

3c Click **Next**.

3d To configure an authentication card, fill in the following:

ID: (Optional) Specify an alphanumeric number that identifies the card. If you need to reference this card outside of the Administration Console, you need to specify a value here. If you do not assign a value, the Identity Server creates one for its internal use.

Text: Specify the text that is displayed on the card to the user

Image: Specify the image to be displayed on the card. Select the image from the drop down list. To add an image to the list, click **Select local image**.

Login URL: (Conditional) If you are configuring an authentication card for SAML 1.1, specify an Intersite Transfer Service URL. For Figure 5-14 on page 375, specify the following value:

```
https://idp.sitea.novell.com:8443/nidp/saml/idpsend?PID=https://
idp.siteb.novell.com:8443/nidp/saml/metadata&TARGET=https://
idp.siteb.novell.com:8443/nidp/app
```

For more information, see “Specifying the Intersite Transfer Service URL for the Login URL Option” on page 166.

Show Card: Determine whether the card is shown to the user. If this option is not selected, the card is only used when a service provider makes a request for the card. For this scenario, select this option.

Passive Authentication Only: Do not select this option.

3e Click **Finish** > **OK**.

3f Update the Identity Server.

Wait for the health status to return to green.

4 Continue with one of the following:

- If you are using Liberty or SAML 2.0, continue with “Verifying the Trust Relationship” on page 381.
- If you are using SAML 1.1, continue with “Configuring SAML 1.1 for Account Federation” on page 384.

Verifying the Trust Relationship

Before continuing with federation configuration, you need to verify that Site A and Site B trust each other.

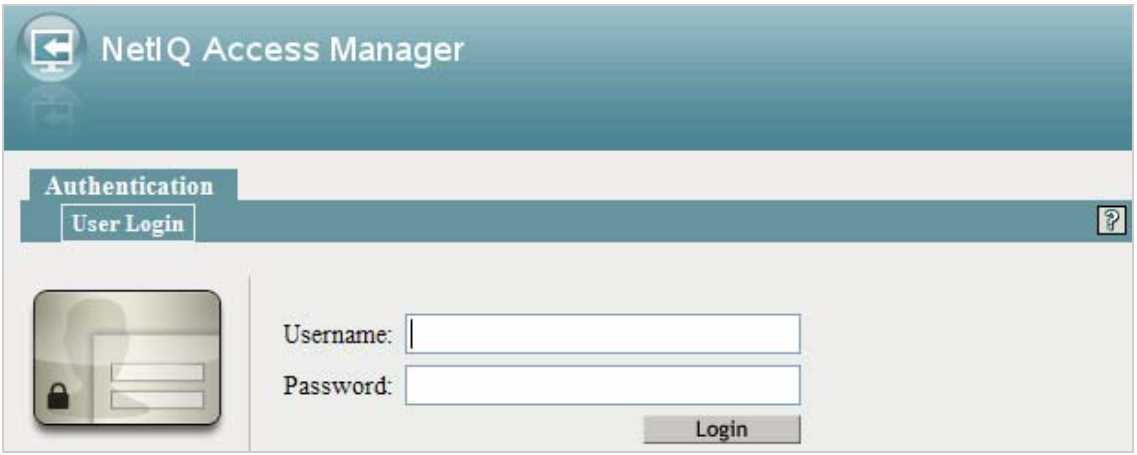
1 To test the trusted relationship, log in to the user portal of Site B. For Site B in Figure 5-15, specify the following:

```
https://idp.siteb.novell.com:8443/nidp/app
```

In this configuration, the customizable image was used for the Liberty authentication card.

2 Click the menu, then click Liberty (or SAML2) authentication card.

You are directed to Site A for login, with the default card selected for you.



3 Enter the credentials for a user from Site A.

The Federation consent prompt appears.

NOTE: To disable this prompt, add the following parameter in the `web.xml` file under the ldapLoadThreshold context parameter:

```
<context-param><param-name>federationConsent</param-name><param-value>true</
param-value></context-param>
```

Linux: `/opt/novell/nids/lib/webapp/WEB-INF/web.xml`

Windows: `\Program Files (x86)\Novell\Tomcat\webapps\nidp\WEBINF\web.xml`

---

4 Click **Yes**.

You are returned to the login page for Site B.

5 Enter the credentials of a user from Site B that you want to federate with the user from Site A.

These two accounts are now federated. You can enter the URL to the user portal on Site A or Site B, and you are granted access without logging in again.

If you log out and log back in, the accounts are still federated, but you might be prompted for login credentials as you access resources on Site A and Site B. To enable a single sign-on experience, the Identity Server at Site A, the Identity Server at Site B, and the protected resources of the Access Gateways must be configured to share a contract.

6 To enable a single sign-on experience, continue with “Configuring User Authentication” on page 382.

Configuring User Authentication

The following instructions describe one way to enable single sign-on to the Identity Servers and Access Gateways in Figure 5-14 on page 375. It explains how to configure all sites to use the same contract. The instructions explain the following tasks:

- Selecting the contract for federation
- Configuring the contract at Site B to allow authentication at Site A
- Configuring Site A so its contract can satisfy the requirements of the contract at Site B
- Configuring Site A and Site B to use this contract as their default contract

To configure the contracts:

1 Log in to the Administration Console for Site B.

2 Configure the authentication request:

2a Click **Devices** > **Identity Servers** > **Edit** > **Liberty** [or **SAML 2.0**] > **[Name of Identity Provider]** > **Authentication Card** > **Authentication Request**.

2b (Liberty) Verify the settings of the following fields:

Allow federation: Make sure this option is selected. If this option is not selected, users cannot federate their accounts at Site A with an account at Site B.

After authentication: Make sure this option is selected. Enabling this option assumes that a user account exists at the service provider and that the account can be associated with a user’s account at the identity provider.

During authentication: Make sure this option is selected. Enabling this option allows federation to occur when the user selects the authentication card of the identity provider.

2c (SAML 2.0) Verify the settings of the following fields:

Persistent: Select this option to set up a persistent relationship between the two accounts.

After authentication: Make sure this option is selected. Enabling this option assumes that a user account exists at the service provider and that the account can be associated with a user’s account at the identity provider after authentication.

During authentication: Make sure this option is selected. Enabling this option allows federation to occur when the user selects the authentication card of the identity provider.

2d For **Requested By**, select **Use Contracts**.

2e (SAML 2.0) For Context Comparison, accept the default value of **Exact**.

2f In the **Authentication contracts** section, select the name of the contract used by the protected resources and move it to the **Contracts** section.

If the contract you require is not in the list, it has not been configured for federation. See step 3.

2g Click **OK**, then update the Identity Server configuration.

3 (Conditional) Configure the contract at Site B to allow federation:

3a Click **Identity Servers** > **Edit** > **Local** > **Contracts**.

3b Record the URI for the contract you are using. This URI needs to exist as a contract on Site A. The name of the contract can be different at each site, but the URI must be the same.

NOTE: If site A only understands authentication class or type, select **Use Types** in the **Requested By** field and specify the authentication class in the **Allowable Class** field. Record the allowable class for the contract you are using. This allowable class must exist as a contract on site B. The name of the contract can be different at each site, but the allowable class must be the same.

3c Click the name of the contract.

3d Make sure the **Satisfiable by External Provider** option is selected.

3e Click **OK** twice, then update the Identity Server if you made any changes.

3f Return to Step 2 to select the contract.

4 Verify that Site A contains the same contract:.

4a Log in to the Administration Console for Site A.

4b Click **Identity Servers** > **Edit** > **Local** > **Contracts**.

4c Match the URI from step 3b to a contract.

NOTE: Match the allowable class if you have selected **Use Types** in the **Requested By** field at site B.

If such a contract does not exist, you need to create it. For help, see "Configuring Authentication Contracts" on page 289.

4d Click **OK**.

5 In the Administration Console for Site A, click **Identity Servers** > **Edit** > **Local** > **Defaults**.

6 For the Authentication Contract, select the name of the contract from step 5c.

7 (Conditional) If you have multiple user stores, set the default contract for each user store.

8 Click **OK**, then update the Identity Server.

9 Test the configuration:

9a Enter the URL to the user portal of Site B.

9b Click the federated login link to Site A.

9c Enter the credentials for Site A and log in.

9d Enter the URL for a protected resource at Site B.

You are granted access without being prompted for credentials.

10 If you want to allow federated users to log in at Site A rather than using the card at Site B to redirect them to Site A, complete the following tasks:

10a In the Administration Console for Site B, click **Devices** > **Identity Servers** > **Edit** > **Local** > **Defaults**.

10b For the Authentication Contract, select the name of the contract whose URI matches the URI of the contract used by Site A.

10c Click **Liberty** [or **SAML 2.0**] > **[Name of Identity Provider]** > **Authentication Card** > **Authentication Request**.

10d In the **Options** section, enable the **Use automatic introduction** option.

This enables single sign-on to Site B when the user has already federated the accounts at the two sites.

10e Click **OK**, then update the Identity Server.

10f To test single sign-on, log in to the user portal on Site A, then enter a URL for a protected resource at Site B.

Configuring SAML 1.1 for Account Federation

SAML 1.1 does not support user-controlled federation, but you can configure it so that accounts that match are automatically federated. The Liberty and SAML 2.0 protocols allow users to federate accounts without sharing any common attributes, but the SAML 1.1 protocol requires that the user accounts need to share some common attributes in order for SAML 1.1 to match them and allow federation.

- "Configuring User Account Matching" on page 384
- "Configuring the Default Contract for Single Sign-On" on page 386
- "Verifying the Trust Relationship with SAML 1.1" on page 386

Configuring User Account Matching

When federating with SAML 1.1, the security of a user matching method depends upon the accuracy of the mapping. You need to select an attribute or attributes that uniquely identify the user at both Site A and Site B. The attributes must identify only one user at Site A and match only one user at Site B. If the attributes match multiple users, you have a security problem,

The following steps use the e-mail address of the user and the LDAP mail attribute to set up a matching rule that matches one user account at Site A with one user account at Site B. To securely use such a matching rule, you need to have a rule in place at both Site A and Site B to ensure that all users have unique e-mail addresses.

Configuring Site B for User Account Matching

1. In the Administration Console of Site B, click **Devices** > **Identity Servers** > **Servers** > **Edit** > **SAML 1.1** > **[Identity Provider]** > **User Identification**.
2. For the **Satisfies contract** option, select the contract that you want to use for single sign-on.

   For this example, select **Secure Name/Password-Form**.
3. Select **Attribute matching**.

   The **Prompt for password on successful match** option is automatically selected. Leave this option enabled.
4. Click the **Define Attribute Matching Settings** icon.
5. Move the user store that you want to search for the attribute to the **User stores** list.
6. For the **User Matching Expression**, select **New User Matching Expression**.
7. Specify a name for the matching expression, such as email.
8. In **Logic Group 1**, click the **Add Attributes** icon, select **Ldap Attribute:mail [LDAP Attribute Profile]**, then click **OK**.

   The form allows you to create a very complex set of matching rules, with multiple conditions. This example uses one attribute, the simplest form of a matching expression.
9. Click **Finish**, then select your matching expression for the **User Matching Expression**.
10. Click **OK**.

11 Click **OK** twice, then update the Identity Server.

12 Continue with "Configuring the Attribute for Sharing" on page 385.

Configuring the Attribute for Sharing

1 In the Administration Console of the Site B (the service provider), click **Devices** > **Identity Servers** > **Shared Settings**.

2 Click **Attribute Sets**, then click **New**.

3 Specify a **Set Name**, such as email, then click **Next**.

4 Click **New**, then fill the **Add Attribute Mapping** options:

Local attribute: Select **Ldap Attribute:mail [LDAP Attribute Profile]**.

Remote attribute: Specify a name, such as email. Make sure you use the same remote name in the mapping for both Site B and Site A.

Leave the other options set to their default values.

5 Click **OK**, then click **Finish**.

Your newly created attribute mapping appears in the list of Attribute Sets.

6 Repeat step1 through step 5 for Site A (the identity provider).

If Site A and Site B are imported into the same Administration Console, skip this step.

7 Continue with "Configuring the Providers to Use the Shared Attribute" on page 385.

Configuring the Providers to Use the Shared Attribute

You need to configure Site A to send the shared attribute with the authentication credentials, and you need to configure Site B to process the shared attribute that is included with the authentication credentials.

1 In the Administration Console for Site B, click **Devices** > **Identity Servers** > **Edit** > **SAML 1.1** > **[Name of Identity Provider]** > **Attributes**.

2 For the **Attribute set**, select the set name you created in "Configuring the Attribute for Sharing" on page 385.

3 Move the email attribute so that it is obtained at authentication.

4 Click **OK** twice, then update the Identity Server.

5 In the Administration Console for Site A, click **Devices** > **Identity Servers** > **Edit** > **SAML 1.1** > **[Name of Service Provider]** > **Attributes**.

6 For the **Attribute set**, select the set name you created in "Configuring the Attribute for Sharing" on page 385.

7 Move the email attribute so that it is sent with authentication.

8 Click **OK** twice, then update the Identity Server.

9 Continue with "Configuring the Default Contract for Single Sign-On" on page 386

Configuring the Default Contract for Single Sign-On

The Identity Servers at Site A and Site B need to use the contract you specified in your user matching expression to be the default contract for Site A, Site B, and the protected resources of the Access Gateway.

For the user matching expression contract, see step 2 in "Configuring Site B for User Account Matching" on page 384.

To configure the default contracts for Site A and Site B:

1. In the Administration Console for Site B, click **Devices** > **Identity Servers** > **Edit** > **Local** > **Defaults**.
2. For the Authentication Contract, select the name of the contract used by the user matching expression.
3. Click **OK**, then update the Identity Server.
4. For Site A, repeat step 1 through step 3.
5. For the Access Gateway, review the contracts you have assigned to the protected resources:
   - **5a** In the Administration Console for Site B, click **Devices** > **Access Gateways** > **Edit** > **[Name of Reverse Proxy]** > **[Name of Proxy Service]** > **Protected Resources**.
   - **5b** For single sign-on, change the contract to match the contract for the user matching expression.
   - **5c** (Conditional) If you have multiple reverse proxies and proxy services, verify the contracts on all protected services that you want enabled for single sign-on.
   - **5d** Click **OK** to save your changes, then update the Access Gateway.
6. Continue with "Verifying the Trust Relationship with SAML 1.1" on page 386.

Verifying the Trust Relationship with SAML 1.1

1. To test the trusted relationship, enter the URL for the user portal of Site B. For Site B in Figure 5-15, you would specify the following:

   ```
   https://idp.siteb.novell.com:8443/nidp/app
   ```

   Use the scroll bar to see all available cards.
2. Click the menu then click card you have configured for SAML 1.1 authentication.

   You are directed to Site A for login.
3. Enter the credentials for Site A.
4. Enter the password for the user at Site B.

   You are directed to the target page specified in the Login URL of the authentication card.

   If you disabled the **Prompt for password on successful match** option on the User Identification page, the accounts are mapped without any user interaction.
5. (Conditional) If you receive an error, try one of the following:
   - If you are not redirected to the target URL on Site B, verify the value you enter for the Login URL option. See Step 3d on page 380.
   - If you receive an authentication error at Site B, verify the user matching setup. See "Configuring User Account Matching" on page 384.
   - If you have enabled logging, open the logging file (`catalina.out` or `stdout.log`) and search for the error string. There must be additional information about the cause of the error in the error string entry as well as log entries before the error sting.
6. (Optional) If your protected resources on Site A and Site B use the same contract, enter the URLs of these resources.

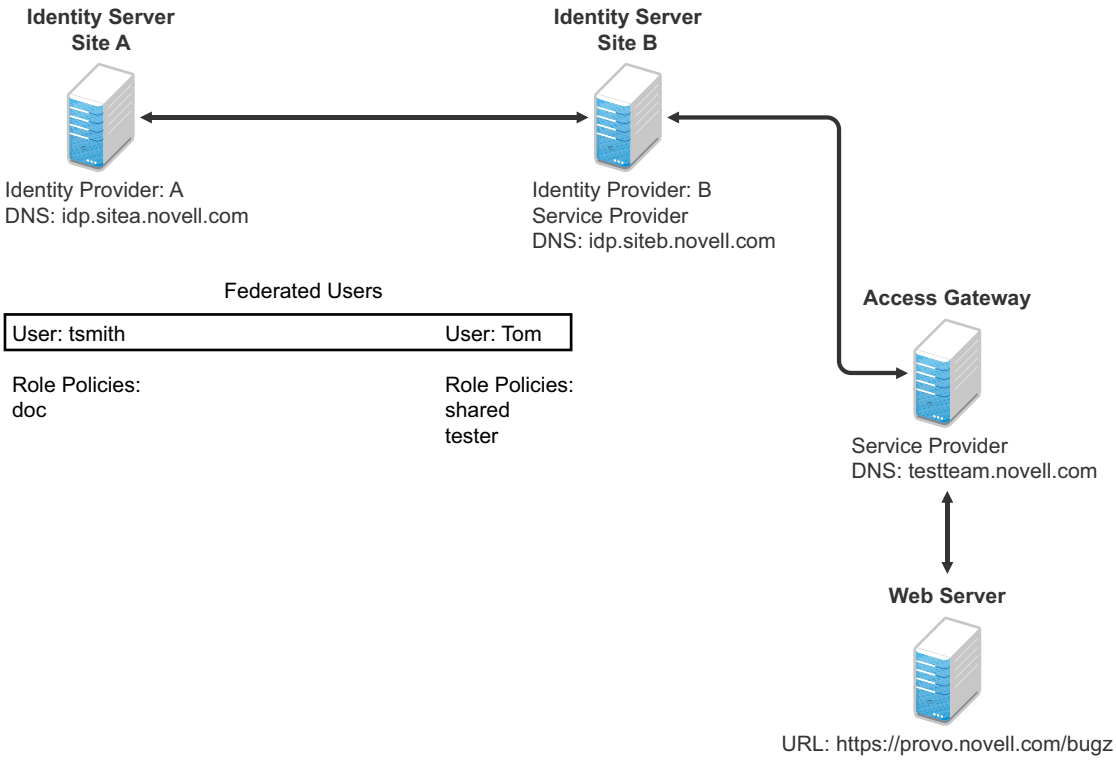
   You are granted access without entering any additional credentials.

Sharing Roles

When two Identity Servers are configured to trust each other, one as an identity provider and the other as a service provider, they can be configured so that roles are shared. The following instructions are written for when both the identity provider and the service provider are NetIQ Identity Servers. If you are using a third-party identity or service providers, you need to modify the instructions.

Figure 5-16 illustrates a configuration where Identity Server of Site A is acting as an identity provider for Site B. When you configure the Identity Servers correctly, the Access Gateway can use the roles defined for the users of Site A in its policies.

Figure 5-16 *Two Federated Identity Servers*



The key to sharing roles is to set up the configuration so that the SAML assertion that the identity provider (Site A) sends to the service provider (Site B) contains the roles that the user has been assigned. Site B evaluates the roles and assigns them to the federated users at Site B. The Access Gateway can use these roles in its policy evaluations, and grant or deny access based on the assigned roles.

For example, when user tsmith authenticates to Site A, tsmith is assigned the role of doc. Tom, a user at Site B, is federated with the tsmith user. The doc role is shared with Site B, and Site B contains a policy that assigns users with the shared doc role to the tester role. The Access Gateway is configured with an Authorization policy that grants access to a resource when the requester is assigned the tester role. However, Tom does not have the qualifications at Site B to be assigned the tester role.

In this scenario, when Tom requests access to the protected resource at Site B, a login page with a federated link to Site A is displayed. If Tom selects to log in to Site A, Site A assigns him to the doc role. The doc role is sent with tsmith's authentication credentials to Site B. Site B evaluates the credentials and assigns Tom to the tester role because the following conditions are met:

- Tom is federated with tsmith.

- tsmith was assigned the doc role.
- The shared role and tester policies on Site B qualify the user to be assigned the tester role.

When the Access Gateway evaluates the credentials of Tom, Tom is granted access to the protected resource because he now has the tester role.

This section describes how to set up such a configuration. It assumes that the following have already been done:

- The trusted relationship between the identity provider and service provider is set up. For configuration instructions, see “Establishing Trust between Providers” on page 378.
- The following policies have been created: the doc role policy at Site A, the tester role policy at Site B, and the Authorization policy (that uses the tester role) for the Access Gateway. The following instructions explain how to set up the shared policy.

This section explains how to configure Site A and Site B so that Site A shares its roles with Site B.

- “Configuring Role Sharing” on page 388
- “Verifying the Configuration” on page 391

Configuring Role Sharing

There are three major tasks for configuring role sharing. You need to configure a shared attribute for transferring the roles. You need to configure the identity provider and the service provider so that the role assignments can be added to the attribute and retrieved from the attribute. Finally, you need to create a shared Role policy for each role sent to the service provider. This policy defines how the role must be processed.

The following sections describe these configuration tasks:

- “Defining a Shared Attribute Set” on page 388
- “Obtaining the Role Assignments” on page 389
- “Configuring Policies to Process Received Roles” on page 389

Defining a Shared Attribute Set

1. In the Administration Console of the Site A (the identity provider), click **Devices** > **Identity Servers** > **Shared Settings**.
2. Click **Attribute Sets**, then **New**.
3. Specify a **Set Name**, such as role_sharing, then click **Next**.
4. Click **New** and fill the **Add Attribute Mapping** options:

   Local attribute: Select **All Roles**.

   Remote attribute: Specify a name, such as roles. Make sure you use the same remote name in the mapping for both the identity provider and the service provider.

   Leave the other options set to their default values.
5. Click **OK**, then click **Finish**.

   Your newly created attribute mapping appears in the list of Attribute Sets.
6. Repeat Step 1 through Step 5 on Site B (the service provider).
7. Continue with “Obtaining the Role Assignments” on page 389.

Obtaining the Role Assignments

1 To export the roles from the identity provider, log in to the Administration Console for the identity provider. (In Figure 5-16, this is Site A.)

   1a Click **Devices** > **Identity Servers** > **Edit** > **Liberty** > **[Name of Service Provider]** > **Attributes**.

   If you are using SAML 2.0 or SAML 1.1 protocol, the steps are the same. You just need to click the appropriate tab after clicking **Edit**. The path is the same for these protocols.

   1b Select the attribute set you created, then move **All Roles** so this attribute is sent with authentication.

   1c Click **OK**.

   1d Update the Identity Server of Site A.

2 To import the roles from the identity provider to the service provider, log in to the Administration Console for the service provider. (In Figure Figure 5-16, this is Site B.)

   2a Click **Devices** > **Identity Servers** > **Edit** > **Liberty** > **[Name of Identity Provider] >** > **Attributes**.

   2b Select the attribute set you created, then move **All Roles** so this attribute is obtained with authentication.

   2c Click **OK**.

   2d Update the Identity Server of Site B.

   2e Continue with "Configuring Policies to Process Received Roles" on page 389.

Configuring Policies to Process Received Roles

For each role that is sent from Site A, you need to create a Role policy that specifies the role that must be activated on Site B. For example, suppose the tsmith user from Site A is assigned the doc role at authentication. You can create a Role policy on Site B that assigns the tester role to anyone with the doc role from Site A.

1 Log in to the Administration Console for Site B.

2 Click **Policies** > **Policies** > **New**.

3 Specify a name for the policy, select **Identity Server: Roles** for the type, then click **OK**.

4 In the **Condition Group 1** section, click **New**, then select **Roles from Identity Provider**.

5 (Conditional) If you have federated with more than one identity provider, select the provider. If you have federated with only one identity provider, the provider is selected for you.

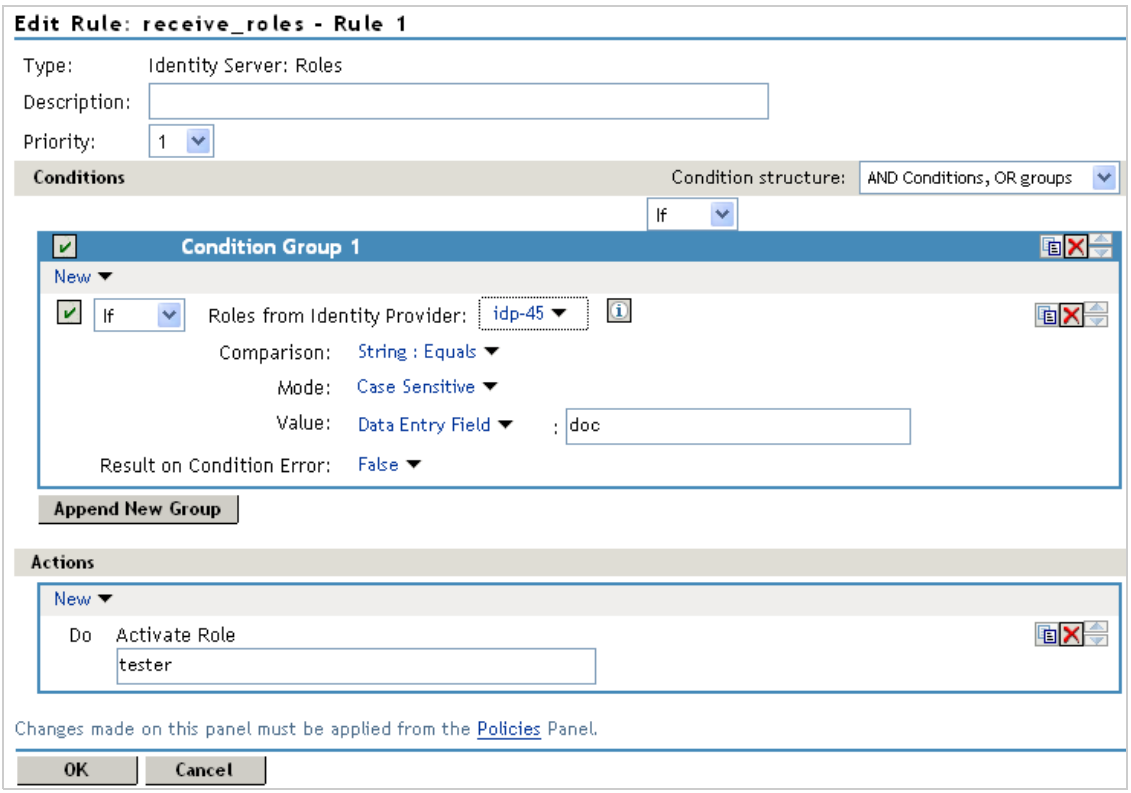
   In this example, you have federated with only the identity provider at Site A, and it is selected for you.

6 For the value, select **Data Entry Field**, then specify the name of a role that is assigned by Site A, for example doc.

   If you leave **Mode** set to **Case Sensitive**, make sure you specify the case correctly.

7 In the **Actions** section, specify the role to activate on Site B for the role received from Site A.

   Your policy must look similar to the following:



8 Click **OK** twice, then click **Apply Changes**.

9 To enable the role for the Identity Server, click **Identity Servers** > **Edit** > **Roles**.

10 Select the role, then click **Enable**.

11 (Optional) Repeat Step 2 through Step 10 for other roles assigned at Site A.

If you have other Role policies at Site A, you need to set up Role policies at Site B to have the roles activated. For example, if Site A had a Tester Role policy and you wanted users assigned to the Tester Role policy to also be assigned to the Tester Role policy at Site B, you could create a separate policy for this activation, or you could add an Or condition group with a value field of tester to the policy in Step 7. The policy would assign federated users who belonged to the doc or tester roles at Site A, to the tester role at Site B.

12 To test role sharing:

12a Enter the URL of a protected resource that requires a role for access. For the policy above, it would be a resource requiring the tester role.

12b Click the federated link to Site A.

12c Log in with the credentials of a user who is assigned the doc role.

You are granted access to the resource. If you are denied access, continue with "Verifying the Configuration" on page 391 to discover the problem.

Verifying the Configuration

This section traces the role assignment from the Identity Server that assigns it to the user, through the Identity Server that receives the roles with the user's authentication assertion, to the policy evaluation. If you are having trouble, this must help you determine the source of the problem.

The following procedures refer to the configuration displayed in Figure 5-16, "Two Federated Identity Servers," on page 387. A tsmith user from Site A, who is assigned the doc role, is federated with a Tom user at Site B. Site B does not assign Tom the tester role. The Web server has been configured to protect the bugz site, which requires the tester role.

To verify the configuration:

1. Make sure policy logging is enabled on the identity provider and the service provider. Make sure that you enable at least Application logging at an Info level.

   For configuration procedures, see "Configuring Logging for Identity Server" on page 852.

   You can access log files for downloading and viewing by clicking **Auditing** > **General Logging**.

2. Have a user access a resource that is protected by a policy requiring a role from Site A.

   For this trace, the tsmith user from Site A requests access to the bugz page. The user uses the federated link and logs in with the credentials of the tsmith user.

3. Verify that Site A is assigning the user the role.

   3a View the `catalina.out` file (Linux) or the `stdout.log` file (Windows) of the Identity Server at Site A.

   3b Search for the name of the role. You must find a line similar to the following:

   ```
   <amLogEntry> 2009-08-22T20:30:19Z INFO NIDS Application: AM#500105013:
   AMDEVICEID#C5F467BA50B009AC: AMAUTHID#DEEF6BEC3655DEB71CA56832DDDF866E:
   Authenticated user cn=tsmith,o=novell in User Store sitea-nids-user-store
   with roles doc,authenticated. </amLogEntry>
   ```

   If the role you need is not listed, look at the policy evaluation trace to discover why the user has not been assigned the role. For more information about how to understand role traces, see "Role Assignment Traces" on page 1036.

4. Verify that Site A is sending an authentication assertion to Site B.

   In the `catalina.out` file (Linux) or the `stdout.log` file (Windows) of the Identity Server from Site A, look for lines similar to the following:

   ```
   <amLogEntry> 2009-08-22T20:30:19Z INFO NIDS Application: AM#500105018:
   AMDEVICEID#C5F467BA50B009AC: AMAUTHID#DEEF6BEC3655DEB71CA56832DDDF866E:
   Responding to AuthnRequest with artifact
   AAPLsCVpfv3ha9Mpn+cUiCXcf3D63sc0QfscL5mZaaygHBKVOOh9aPSQ </amLogEntry>

   <amLogEntry> 2009-08-22T20:30:19Z INFO NIDS Application: AM#500105019:
   AMDEVICEID#C5F467BA50B009AC: AMAUTHID#F8B1C147EB3DDEFE9A3DB0827BA8E4A3:
   Sending AuthnResponse in response to artifact
   AAPLsCVpfv3ha9Mpn+cUiCXcf3D63sc0QfscL5mZaaygHBKVOOh9aPSQ </amLogEntry>
   ```

   If you do not see these types of entries, verify that you have configured Site A to send the roles. See "Obtaining the Role Assignments" on page 389.

5. Verify that Site B is receiving the SAML assertion with the roles.

   In the `catalina.out` file (Linux) or the `stdout.log` file (Windows) of the Identity Server from Site B, look for lines similar to the following:

```
<amLogEntry> 2009-08-22T20:30:19Z INFO NIDS Application: AM#500105020:
AMDEVICEID#488475009C6D3DDF: AMAUTHID#0FBA0CF7E41E6C7F9121DABB918D34F4:
Received and processing artifact from IDP -
AAPLsCVpfv3ha9Mpn+cUiCXcf3D63sc0QfscL5mZaaygHBKVOOh9aPSQ </amLogEntry>

<amLogEntry> 2009-08-22T20:30:19Z INFO NIDS Application: AM#500105021:
AMDEVICEID#488475009C6D3DDF: AMAUTHID#0FBA0CF7E41E6C7F9121DABB918D34F4:
Sending artifact AAPLsCVpfv3ha9Mpn+cUiCXcf3D63sc0QfscL5mZaaygHBKVOOh9aPSQ to
URL https://rholm.provo.novell.com:8443/nidp/idff/soap at IDP </amLogEntry>
```

The artifact ID must be the same as the artifact ID in Step 4.

If you do not see these types of entries, verify that you have configured Site B to receive the roles. See “Obtaining the Role Assignments” on page 389.

6 Verify that Site B is evaluating the received role assignments and activating the roles.

In the `catalina.out` file (Linux) or the `stdout.log` file (Windows) of the Identity Server from Site B, search for a policy evaluation for RolesFromIdentityProvider. You must find lines similar to the following:

```
~~CO~1~RolesFromIdentityProvider(6670):https://ipd.sitea.provo.novell.com:
8443/nidp/idff/metadata:TESTER,DOC,AUTHENTICATED~com.novell.nxpe.condition.
NxpeOperator@string-equals~(0):hidden-param:hidden-value:~~~True(69)

~~PA~ActionID_1203705845727~~AddRole~tester~~~Success(0)

<amLogEntry> 2009-08-22T20:30:20Z INFO NIDS Application: AM#500105013:
AMDEVICEID#488475009C6D3DDF: AMAUTHID#0FBA0CF7E41E6C7F9121DABB918D34F4:
Authenticated user cn=Tom,o=novell in User Store Internal with roles
tester,authenticated. </amLogEntry>
```

The policy evaluation shows that the condition evaluates to true and that the tester role is activated. Tom is the user that is federated with the tsmith user, and the entry shows that Tom has been assigned the tester role.

If you do not see a policy evaluation for RolesFromIdentityProvider, make sure you have created such a Role policy and that you have enabled it. See “Configuring Policies to Process Received Roles” on page 389.

7 If the use has been assigned the correct role, the last step is to verify how the embedded service provider evaluated the policy protecting the resource.

In the `catatina.out` file of the `ipd-esp` file for the Access Gateway, search for lines similar to the following for the authorization policy trace:

```
<amLogEntry> 2009-08-22T20:30:20Z INFO NIDS Application: AM#501102050:
AMDEVICEID#esp-2559E77C93738D15: AMAUTHID#BCF3CB40B51E8A0AF8582BEF762B4DDD:
PolicyID#65LN233O-KN19-1L7M-176M-P942LMN6P832: NXPESID#1411: AGAuthorization
Policy Trace:
 ~~RL~1~~~~Rule Count: 2~~Success(0)
 ~~RU~RuleID_1198874340999~Allow_Tester~DNF~~1:1~~Success(0)
 ~~CS~1~~ANDs~~1~~True(69)
 ~~CO~1~CurrentRoles(6660):no-param:TESTER,AUTHENTICATED~com.
novell.nxpe.condition.NxpeOperator@string-substring~SelectedRole
(6661):hidden-param:hidden-value:~~~True(69)
 ~~PA~1~~Permit Access~~~~Success(0)
 ~~PC~1~~Document=(ou=xpemlPEP,ou=mastercdn,ou=ContentPublisher
Container,ou=Partition,ou=PartitionsContainer,ou=VCDN_Root,ou=accessManagerCon
tainer,o=novell:romaContentCollectionXMLDoc),Policy=(Allow_Tester),Rule=(1::Ru
leID_1198874340999),Action=(Permit::1)~~~~Success(0)
 </amLogEntry>
```

If the PA line does not evaluate to Permit Access, then you need to review the Authorization policy and discover the conditions, other than the tester role, that must be met to permit access.

Setting Up Federation with Third-Party Providers

Setting up federation with providers other than NetIQ Identity Servers requires the same basic tasks as setting up federation with NetIQ Identity Servers, with some modifications.

When you set up federation with identity providers and service providers that are controlled by a single company, you have access to the Administration Consoles for both Identity Servers and know the admin credentials. When setting up federation with another company, additional steps are required.

- You need to negotiate with the other company and gain approval for federation because metadata must be shared and both sites require configuration. You need to negotiate a schedule for these configuration changes.
- The other site might not be using Access Manager for its identity or service provider. The basic tasks need to be modified to accommodate how that implementation shares metadata, authentication methods, and roles.
- Many SAML 1.1 providers do not support a metadata URL, and the data must be imported manually.

  For example, instead of sharing URLs that allow you to import metadata, you might need to share the actual metadata and paste it into the configuration. The NetIQ Identity Server validates the metadata of another identity provider or service provider; some implementations do not validate it. If the Identity Server determines that the metadata is invalid, you need to negotiate with the provider to send you metadata that has been validated.
- Most third-party providers do not support authentication cards and contracts. However, most do support either authentication types or authentication URIs. You can use either of these to map from their authentication procedure to an Identity Server authentication contract.

For sample implementations with third-party providers that explain the modifications that were required to set up the federation, see the following:

- “Integrating Novell's Access Manager with Shibboleth's IDP Server” (http://www.novell.com/communities/node/6943/integrating-novells-access-manager-shibboleths-idp-server)
- “Integrating Google Apps and Novell Access Manager using SAML2” (http://www.novell.com/communities/node/8645/integrating-google-apps-and-novell-access-manager-using-saml2)
- “SAML 1.1 with Concur” (http://www.novell.com/coolsolutions/appnote/19673.html)

Service Provider Brokering

The Service Provider Brokering (SP Brokering) feature enables the Identity Server to act as a federation gateway or a service provider broker. This federation gateway allows you to connect to different protocols such as Liberty, SAML 1.1, and SAML 2.0. You can use SP Brokering with the Intersite Transfer service of the identity provider. Intersite Transfer service enables authentication at a trusted service provider.SP Brokering helps companies establish trust between identity providers and their service providers that support different federation protocols. For example, an identity provider that supports SAML 2.0 can provide authentication to a Liberty or SAML 1.1 service provider by using SP broker.

SP Brokering helps reduce the number of trust relationships between an identity provider and their service provider. For example, identity providers can now provide authentication to their service providers by establishing a single trust relationship instead of multiple trust relationships. Similarly, a service provider must establish a single trust relationship with SP Broker to receive authentication from several identity providers.

You can control the authentication flow between several identity providers and service providers in a federation circle by allowing the administrator to configure policies that control Intersite Transfers. For example, an administrator can configure a policy with SP Broker that allows only certain users from an identity provider to be authenticated at a given service provider.

An Intersite Transfer URL has the following format: `https://<identity provider>/idpsend?PID=<Service Provider ID>&TARGET=<final_destination_URL>`

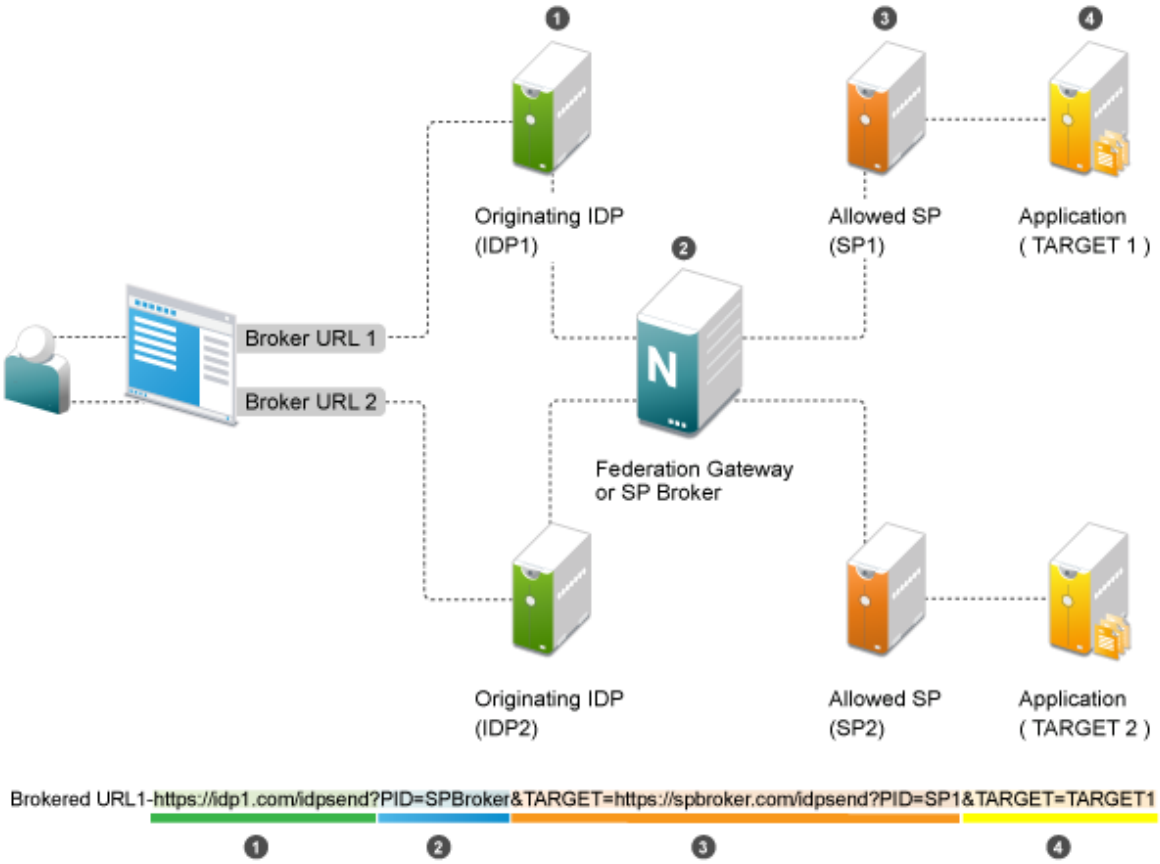
This Intersite Transfer URL consists of three parts:

- `https://<identity provider>:` The user can authenticate at the identity provider.
- `/idpsend?PID=<Service Provider ID>`: Authentication occurs at the service provider represented by the service provider ID at the identity provider.
- `&TARGET=<final_destination_URL`: The user is finally redirected to the specified target URL associated with the service provider.

A Web page is created with many Intersite Transfer URLs for each combination of identity provider, service provider, and the target application.

For more information about the Intersite Transfer Service, see “Using the Intersite Transfer Service” on page 164.

This following illustration explains the flow of providing access to the target URL by using SP Brokering:



Web Page (User Portal): A Web page (user portal) is created with a list of URLs called Brokered URLs, which provide access to various target applications.

Originating Identity Providers: The Originating Identity Provider is the identity provider with which the user credentials are stored for authentication. The Origin IDP must be configured as a Liberty/SAML1.1/SAML2.0 trusted identity provider in the SP Broker.

Federation Gateway or SP Broker: The Federation Gateway or SP Broker is a NetIQ identity provider that can be configured to control the authentication between several Origin IDPs and Allowed SPs in a federation circle.

Allowed Service Provider: The Allowed SP is the service provider in which the SP Broker provides authentication. The allowed SP must be configured as a Liberty/SAML1.1/SAML2.0 trusted service provider on SP Broker.

Target Application: The target application is the application running on a Web Sever that is protected by the service provider.

Broker URL: A Broker URL is a specially designed Intersite Transfer URL, which consists of four parts. You can click the brokered URL, which results in the following:

1. You must authenticate with the Originating IDP (`https://idp1.com/idpsend`).
2. The Origin IDP causes an authentication to occur at the SP Broker (`?PID=SPBroker`).

3. The SP Broker causes an authentication to occur at the allowed SP (TARGET=`https://spbroker.com/idpsend?PID=SP1`).
4. You are redirected to the target application (?TARGET=TARGET1).

SP Brokering requests are the Intersite Transfers resulting from brokered URLs processed on the SP Broker. The SP Broker can control the brokering requests before providing an authentication to the service provider. The SP Broker enforces the policies configured by the administrator by either causing the authentication at the service provider or by denying the request.

The SP Broker provides the following options to configure policies that control SP brokering requests:

1. A set of SAML 1.1, SAML 2.0 and Liberty trusted identity providers and trusted service providers can be configured as a brokering group. The brokering request is allowed only if the Origin identity provider and Allowed service provider belong to the same brokering group. Brokering Request is not allowed from an Origin identity provider of one group to an Allowed service provider of another group.
2. In a brokering group, a set of brokering rules can be configured that provides granular control on the brokering requests. For example, a brokering rule can be configured to deny a brokering request from an Origin identity provider to an Allowed service provider, if the user satisfies a certain condition at the SP Broker.

SP brokering is enabled on the Identity Server only if at least one brokering group is enabled. If an Intersite Transfer request is received with neither the origin identity provider nor the Allowed service provider in any of the brokering group, the request is treated as a regular Intersite Transfer and SP brokering controls are not applied.

This chapter provides information about configuring the Access Manager SP Brokering functionalities, various deployment scenarios, and associated configuration details.

- “Functionalities” on page 396
- “Brokering Flow” on page 397
- “Deployment Scenarios” on page 400
- “Configuring a Brokering for Authorization of Service Providers” on page 404
- “Creating and Viewing Brokering Groups” on page 405
- “Generating the Brokering URLs by Using an ID and Target in the Intersite Transfer Service” on page 411
- “Transient Federation within SAML 2.0” on page 412
- “Assigning the Roles for the Origin IDP users in SP Broker Using the Transient Federation Attributes” on page 412
- “Assigning The Local Roles Based On Remote Roles And Attributes” on page 413
- “SP Brokering Example” on page 414

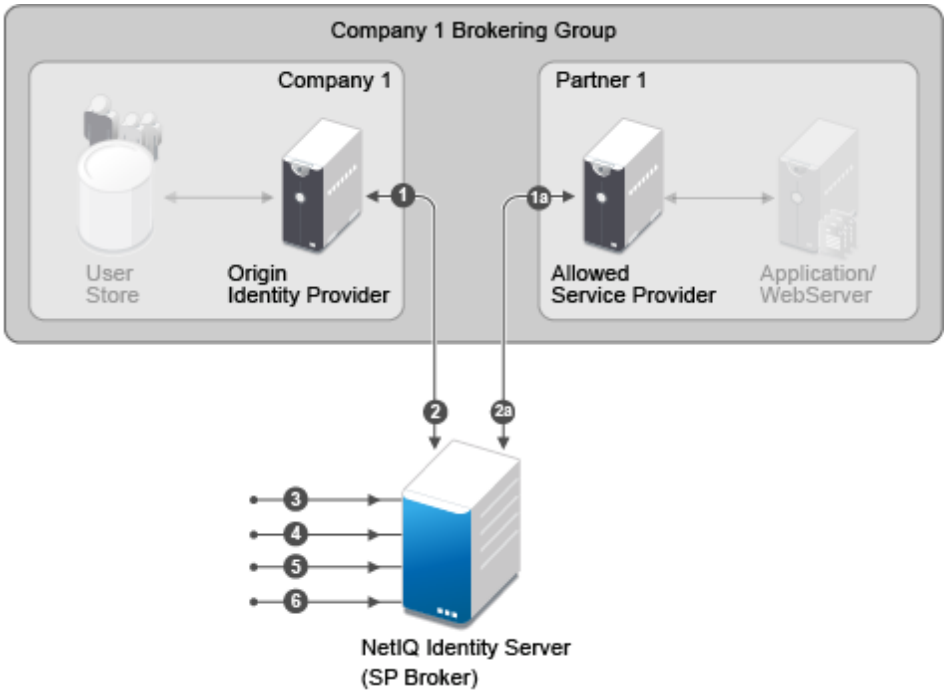
Functionalities

- Defines logical groups for Brokering
  - Brokering happens only among the group members. For example, Brokering of User Group1 users to Application 2 is not allowed.
  - A trusted provider is present in more than one group. For example, common partner is configured as a trusted service provider in the broker. The common partner is part of both Broker Group-1 and Broker Group-2.

- All the brokering rules apply within a group.
  - The brokering rules defines the origin Identity Server, Service Provider and the application target.
  - The brokering rule is attached to *any* role or a specific Identity Server role is defined at Broker Identity Server.
  - The brokering rules are based on prioritized list.

Brokering Flow

Figure 5-17 *Brokering Group Configuration*



The Brokering Group configuration image provides information about how the Identity Provider Brokering group is configured with Service Provider Brokering Group.

1. Identify the Company and Partners' Identity Providers.
   - Company 1 Brokering Group is configured with their Identity Server.
   - *1a* is the partner of Company 1 Brokering Group configured with Service Provider Brokering Group that is Novell Identity Server.
2. The federation is established between the company and partners' Identity Provider and the Service Provider Brokering Group that is Novell Identity Server.
   - Company 1 Brokering Group is configured with their Service Provider Brokering Group that is Novell Identity Server.
   - *2a* is the partner of Company 1 Brokering Group configured with Service Provider Brokering Group that is Novell Identity Server.
3. Create a new brokering group.

   The Service Provider manages the brokering group based on roles.
   - Roles based on Identity Provider authentication.

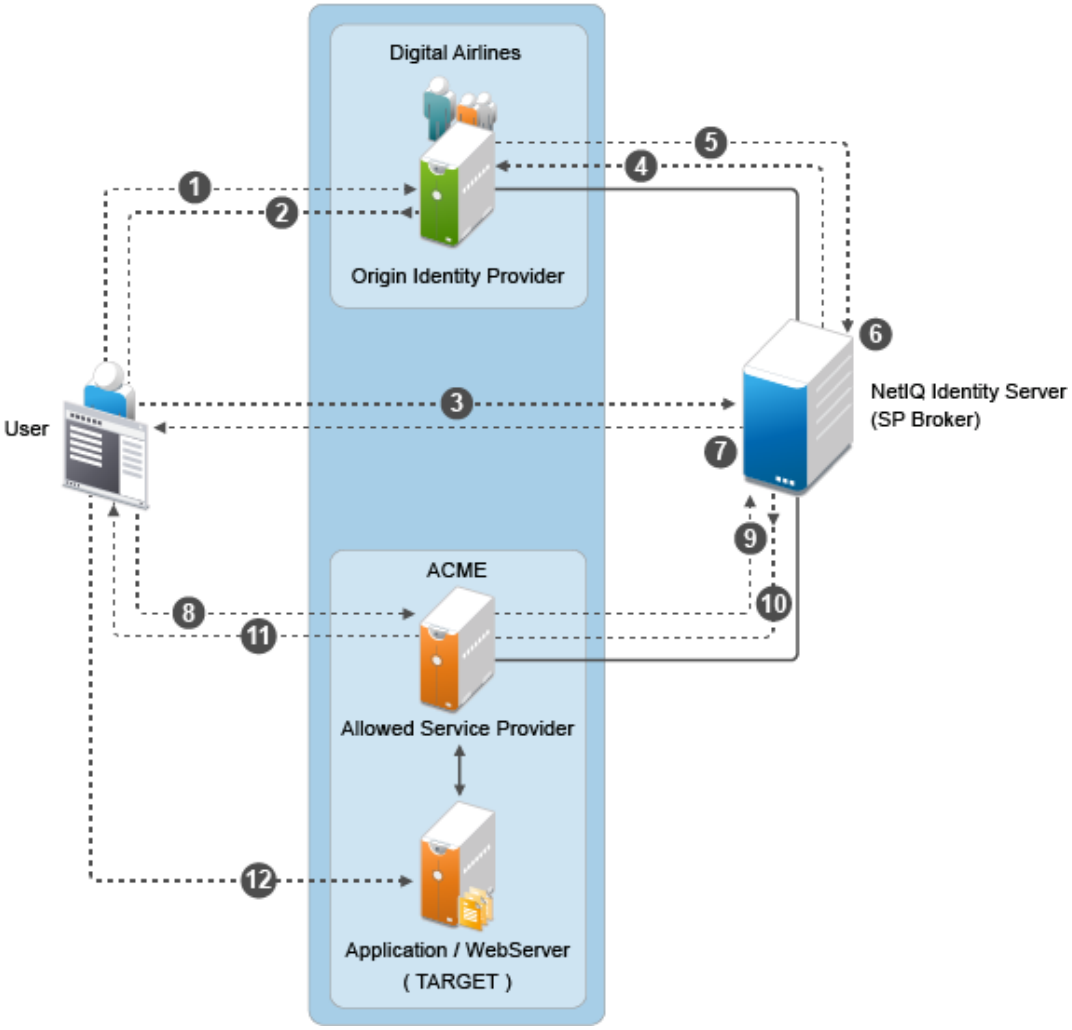
- Roles based on Service Provider brokering authentication.
- Assign the Identity Providers and Service Providers.

4 Using Liberty, SAML 1.1, and SAML 2.0 protocols define policies and do the intersite transfer around the Service Provider Brokering feature.

5 Using the Brokering Service construct URLs.

6 Construct URL for each Identity Provider and Service Provider pair.

Figure 5-18 *Brokering Group Flow*



The Identity Server is being shared to provide Service Provider brokering to a set of logical customers. Company 1 has one partner. All the trusted providers are configured at one broker Identity Server

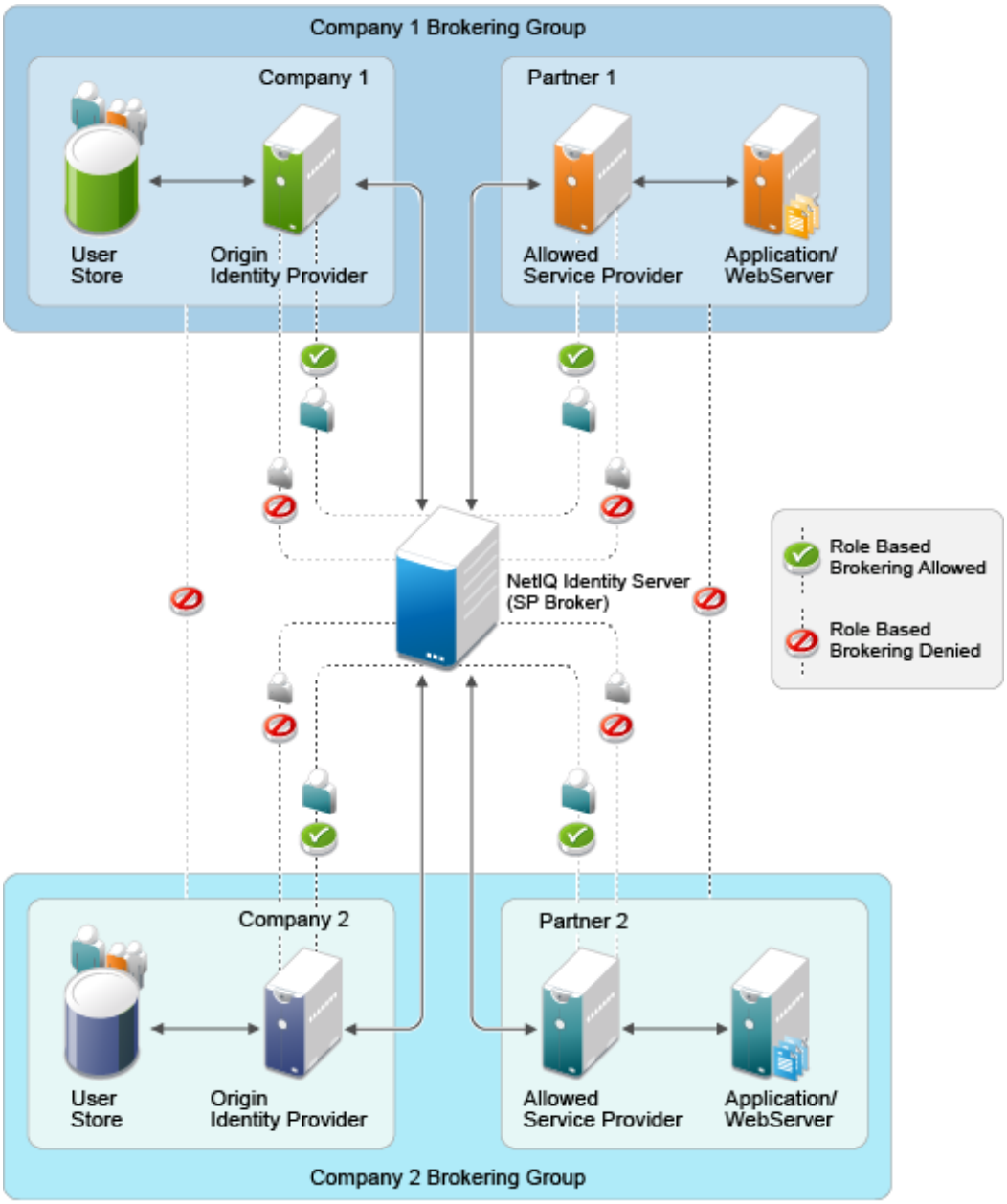
1. User clicks on URL1. The browser send a request to https://idp.customer1.com/nidp/saml2/idpsend?PID=https://brokeridp.verizon.com/nidp/saml2/metadata&ID=partner1-sp-id&TARGET=https://www.partnerapp.partner1.com
2. Customer Identity Provider prompts the user for credentials if not already logged in. User logs in at Customer-1-IDP. The Identity Provider then performs an inter site transfer to Identity Provider Broker. This involves creating an sp-assertion-consumer-URL request and redirects the user to the following URL which eventually lands at Broker Identity Providers' Assertion Consumer URL https://brokeridp.abc.com/nidp/saml2/sp_assertion_consumer
3. POST contents will include SAML Artifact = <artifact> and RelayState=https://brokeridp.abc.com/nidp/saml2/?idpsend=partner1-sp-id&TARGET=https://www.partnerapp.partner1.com
4. The service provider assertion consumer URL processing includes a hook to enforce broker rules.
   - From the Artifact, it finds the trusted provider that it is receiving the artifact from origin trusted provider.
   - If the RelayState contains IDPsend, then it finds the target trusted provider from the RelayState and also finds the target.
   - Using origin trusted provider, the group to which this brokering request belongs is found and a search is made for the policies representing origin trusted provider, target trusted provider and brokering service provider.
   - At this time, only role is unknown. A decision can be taken if the brokering is allowed between origin trusted provider and target tested provider for a particular target or not. If it is allowed then it is proceeded to the next step of artifact resolution.
   - after this request needs further processing of role enforcement which will be known only after an assertion is received from customer identity provider, a flag is set on the Novell identity provider session object. This flag (Broker_role_enforcement) is checked during assertion processing.
5. Artifact resolution happens at customer identity server.
6. Artifact resolution response is sent to the broker identity server which contains the assertion.
7. A new hook is made in the assertion processing.
   - If Broker_role_enforcement flag is set on the session, then Roles are identified for this userBroker rules are again enforced for the Roles.
   - If the brokering is not allowed for the Role an error message is displayed at the browser. Otherwise the browser is redirected back to the Broker Identity Server (to itself) with the following URL https://brokeridp.verizon.com/nidp/saml2/?idpsend=partner1-sp-id&TARGET=https://www.partnerapp.partner1.com
   - Intersite transfer is now made to the DSP with the following URL https://partner.idp.com/nidp/saml2/spassertion_consumer
   - The POST message contains SAML Artifact and RelayState (which contains the target URL).
8. The partner service provider verifies the artifact over SOAP back channel with broker identity servers.
9. Broker Identity Servers resolves the artifact and sends the assertion.

10 Partner Service Providers redirects the browser to the target URL (https://www.partnerapp.partner1.com). It sets its cookie on the browser during the redirection. At this time the user has a valid authenticated session on Partner Service Provider.

11 The Partnerapp.partner1.com validates the session and provides access to the user.

Deployment Scenarios

- “Configuring Trusted Providers at One Broker Identity Server” on page 401
- “Brokering Across Group is not Allowed” on page 401
- “Brokering Within Group is Allowed” on page 401
- “Brokering Within a Group Based On Groups and Members” on page 401



Configuring Trusted Providers at One Broker Identity Server

The Identity Server is shared among two sets of logical customers to provide Service Provider brokering feature.

- The Company 1 Brokering Group consists of Company 1 and Partner 1 logical customers.
- The Company 2 Brokering Group consists of Company 2 and Partner 2 logical customers.

Brokering Across Group is not Allowed

The brokering feature is not allowed among different company groups.

The brokering is not allowed between the logical customers of Company 1 Brokering Group and Company 2 Brokering Group.

Brokering Within Group is Allowed

The brokering feature is allowed among different partners of the company group.

Brokering is allowed between the brokering groups such as COmpany 1 Brokering Group and Company 2 Brokering Group.

- Role based brokering is allowed among Company 1 and Partner 1 logical customers.
- Role based brokering is allowed among Company 2 and Partner 2 logical customers.

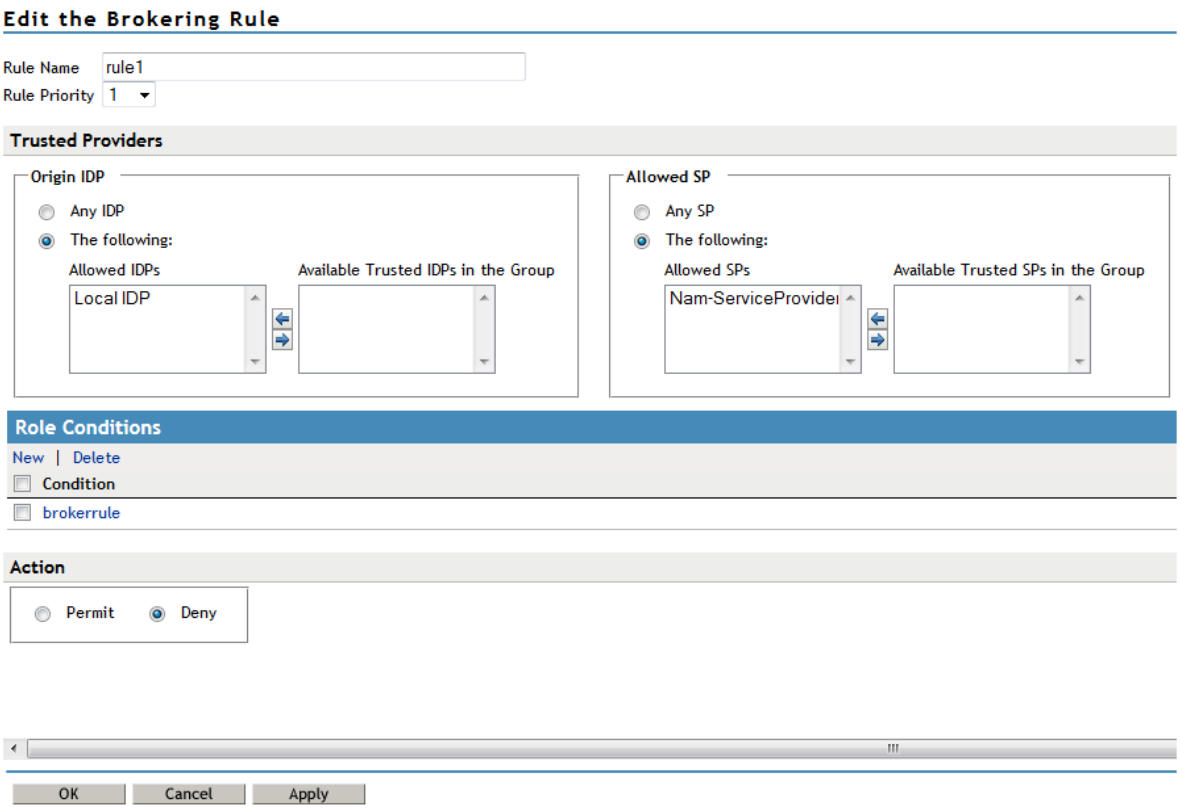
Brokering Within a Group Based On Groups and Members

The brokering feature is allowed among different partners based on roles and groups authentication of the company.

Configuring a Brokering for Authorization of Service Providers

Authorization rules for authorizing service provider requests must be configured from the Access Manager Brokering page. To configure authorization policy, configure the broker rule policy. Ensure that the service providers are configured to the local Identity Server that will be evaluated during authorization. Figure 5-19 on page 405 displays the sample configuration.

Figure 5-19 *SAMl2 Service Provider Initiated Authorization Rule Configuration*



Creating and Viewing Brokering Groups

The identity server cluster configuration provides a **Brokering** tab that you can use to configure the groups and generate brokered URLs.

1. In the Administration Console, click **Devices > Identity Servers > Brokering**.
2. The **Brokering** tab allows you to create new Groups as well as display the configured Groups.The Display Brokering Groups page displays the list of groups configured.

   You can also create, delete, enable, and disable the brokering group on this page.
3. The Display Brokering Groups page displays the following information for each group:

   Group Name: Specifies a unique name to identify the group. When you click on the hyperlink, you can view the Group Details page, where the Group configuration such as name and list of Identity Providers and Service Providers can be modified.

   Enabled: A check mark indicates that brokering is enabled for the group by applying theconfigured rules. A blank means that brokering is disabled.

   Identity Providers: Display the total number of Liberty/SAML1.1/SAML2 IDPs assigned to this group.

   Service Providers: Display the total number of Liberty/SAML1.1/SAML2 SPs assigned to this group.

Brokering Rules: If the rules are not configured, then “No Rules Config” is displayed. The default rule allows for brokering between any IDP to any SP in the group. If new rules are configured, then the first rule name is displayed along with the count of total rules.

- “Creating a Brokering Group” on page 406
- “Configuring Trusted Identity Providers and Service Providers” on page 407
- “Configuring Brokering Rules” on page 407
- “Constructing Brokering URLs” on page 409
- “Validating Brokering Rules” on page 410

Creating a Brokering Group

You can create Broker Group and configure rules for the selected groups. Enter the name of the group and select the trusted providers using the arrow navigation button.

To create a new broker group follow these steps:

1. In the Administration Console, click **Devices > Identity Servers > Brokering**.
2. Click **New**. The Creating Brokering Group page displays.
3. Perform the following actions in the fields:

   Display Name: Specify the brokering group display name.

   Selected IDPs: Select at least one trusted IDP using navigation button.

   Selected SPs: Select at least one trusted SP using navigation button.

   Available Trusted IDPs: Displays Liberty/SAML1.1/SAML2.0 trusted IDP configured on the given IDP cluster (idp_cluster1).

   Available Trusted SPs: Displays Liberty/SAML1.1/SAML2.0 Trusted Service Providers configured on the given Identity Provider Cluster (idp_cluster1).
4. Click **Finish** to complete creation of the brokering group creation.

Configuring Trusted Identity Providers and Service Providers

You can configure the rules between the trusted identity providers and service providers by configuring rules, roles, and actions. You can view the configured rules, create new, delete the existing rule, edit the rules, enable and disable the configured rules.

You can configure the service providers and identity providers for all of the protocols in the Identity Server, which are configured in the Identity Server cluster. Using the brokering group, you can view the list of available service providers and identity providers in the selection box. Using the arrow keys, configure the trusted identity providers and trusted service providers for the respective brokering group.

1. In the Administration Console, click **Devices > Identity Servers** > **Brokering Group Name**. The Configuration page displays the **Trusted Providers, Brokering rules, Construct URL** and **Rule Validation** tabs.
2. Click **Trusted Providers** tab.
3. Specify the display name and configure the brokering groups.

   Display Name: Specify the display name of the configuring brokering group.

   Select IDPs: Configure the selected identity providers using the arrow keys from the available trusted IDPs.

Available Trusted IDPs: Configure the available trusted identity providers using the arrow keys from **Selected Identity Providers** selection box.

Selected SPs: Configure the selected service providers using the arrow keys from the **Available Trusted Service Providers** selection box.

Available Trusted SPs: Configure the available trusted service providers using the arrow keys from the **Selected Service Providers** selection box.

4 Click **OK** to continue and the configured service providers and identity providers details are displayed in the Brokering page.

5 Click **Finish** to complete the rules configuration for the brokering group.

6 Click **Apply** to see the configuration changes.

NOTE: When you log out from the Access Gateway device, then the logout is not propagated on the other Identity Servers if you have SAML 1.1 as one of the trusted provider in the brokering group.

Configuring Brokering Rules

You can create, edit, delete, enable, and the disable brokering rules.

1 In the Administration Console, click **Devices > Identity Servers** > **Brokering**.

2 Click the existing or newly created Brokering Group hyperlink.

3 Click **Rules**. The Brokering Group Rules page is displayed.

Name: Displays the rule name of the brokering group.

Enabled: Displays the status of the brokering group rule.

Identity Providers: Displays the number of identity providers configured to the brokering group.

Service Providers: Displays the number of service providers configured to the brokering group.

Priority: Displays the brokering group rule priority number.

Actions: Displays the configured brokering group rule action status either as permit or deny.

Role Conditions: Displays the brokering group role condition, such as manager and emplyee , configured on the rule page.

4 Click **OK** to continue and display the configured brokering group rule details on the Brokering Rules page.

5 Click **Apply** to see the brokering rule configuration changes.

Creating a Brokering Rule

You can configure the rules to the created brokering groups.

1 In the Administration Console, click **Devices > Identity Servers > Brokering**.

2 Click the existing or newly created Brokering Group hyperlink.

3 Click **Rules**. The Creating Brokering Group page displays.

Rule Name: Specify the name of the rule.

Rule Priority: Select the rule priority from the drop-down list.

NOTE: The default rule specified during creation of the group has a priority of 1. Additional rules can be added, and existing rules can be deleted or modified. You can use the Edit Rules Page to modify the priority of the rules.

Origin IDP: Displays all Identity Servers or one or more Identity Servers that are available in the group.

Allowed SP: Displays all service providers or one or more service providers that are available in the group.

Role Conditions: Displays the brokering group role condition such as manager and employee, configured on the rule page.

Actions: Select the Permit or Deny action radio button for the rule you configure to the brokering group.

NOTE: By default, Access Manager allows any role. If you want to allow access to only particular roles, configure a permit condition for roles with higher priority and configure a deny condition in which no roles are defined with lower priority.

---

4 Click **Finish** to complete configuration of rules for the brokering group.

Deleting a Brokering Rule

1 In the Administration Console, click **Devices > Identity Servers > Edit > Brokering > (Brokering Group in the Brokering Group list) > Rules**.

2 Select the check box of the brokering group rule you want to delete, then click **Delete**. A message is displayed as “Delete selected brokering rule(s)?”.

3 Click **OK** to continue.

Enabling a Brokering Rule

1 In the Administration Console, click **Devices > Identity Servers > Edit > Brokering** > **(Brokering Group in the Brokering Group list)** > **Rules**.

2 Select the check box of the brokering group rule you want to enable.

3 Click **Enable**.The selected brokering group is enabled.

Disabling a Brokering Rule

1 In the Administration Console, click **Devices > Identity Servers > Edit** > **Brokering** > **(Brokering Group in the Brokering Group list)** > **Rules**.

2 Select the check box of the brokering group you want to disable from the brokering group rule configuration.

3 Click **Disable**. The selected brokering group is disabled.

Editing Brokering Rules

You can edit the group rules in the Brokering page.

1 In the Administration Console, click **Devices > Identity Servers > Edit > Brokering**.

2 Click the existing or newly created brokering group hyperlink.

3 Click **Rules** tab.

4 Click the Brokering Rules hyperlink to edit the information. The Edit Brokering Rule page displays the information. You can also edit the information.

You can edit all the fields and modify the information about the Create Brokering Rule page. For more information about create brokering rule, see “Creating a Brokering Rule” on page 408

Constructing Brokering URLs

The Construct URL page helps you to create a URL, which you use in your application to navigate to your trusted partners.

You can generate the URL according to the origin and allowed service provider Identity Servers.

1. In the Administration Console, click **Devices > Identity Servers > Brokering**.
2. Click the existing or newly created brokering group hyperlink.
3. Click **Construct URL**.

   IDP Type: Select the Identity Provider type from the drop-down list. The three types of IDP in the drop-down list are Local IDP, NetIQ IDP, and Other IDP. If you select NetIQ IDP as the IDP type, then you can select the Origin IDP from the drop-down list. If you select Other IDP as the IDP type, you can enter the Origin IDP URL and you can select the Origin IDP from the drop-down list.

   Origin IDP: The Origin identity providers are the trusted providers. The drop-down list displays all the trusted providers created for the specific NetIQ brokering group. Select the Origin IDP from the drop-down list.

   NOTE: If the Origin IDP drop-down list does not list any trusted providers, it is because a local Identity Server exists as a trusted provider. To resolve this, add another Identity Server to the NetIQ brokering group

   ---

   Origin IDP URL: If you select Other IDP as the IDP type, you can enter the Origin IDP URL manually. The <OriginIDPURL> represents (protocol :// domain : port / path ? querystring).

   Provider Parameter Name: If you select Other IDP as the IDP Type, you can enter the trusted provider parameter ID. For more information about Intersite Transfer Service target for a service provider, see “Configuring an Intersite Transfer Service Target for a Service Provider” on page 169

   Target Parameter Name: If you select Other IDP as the IDP type, you can enter the target provider parameter name manually.

   Allowed SP: The allowed service providers are the selected service providers of the trusted roviders. The drop-down list displays all the service providers created for the specific brokering group. Select the service providers from the drop-down list.

   Target URL: Specify the target URL for the specific trusted providers and service provider pair. This URL will be appended to the login URL. Click **Generate** to generate the login URL

   Login URL: The login URL consists of Origin IDP URL and the target URL.
4. Click **Cancel** to close the Construct URL page.

Validating Brokering Rules

The rule validation page helps you to validate the Origin identity providers and the allowed service provider rule according to the role associated with the respective trusted partners.

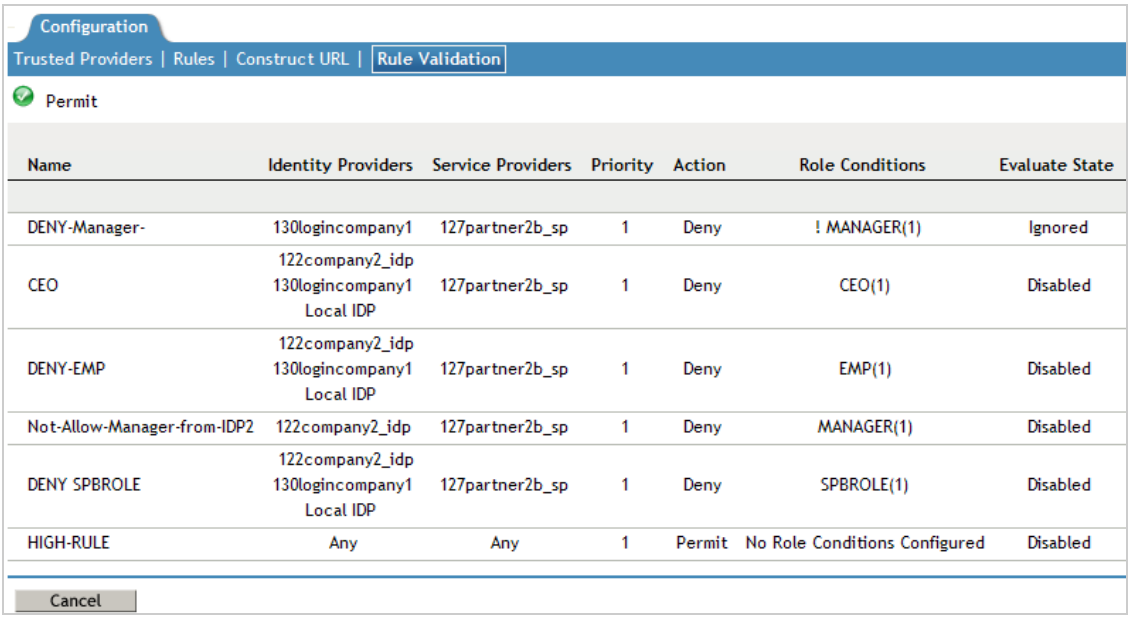
1. In the Administration Console, click **Devices > Identity Servers > Brokering**.
2. Click on the existing or newly created brokering group hyperlink.
3. Click the **Rule Validation** tab.

   Origin IDP: The Origin identity providers are the trusted providers. The drop-down list displays all the trusted providers created for the specific NetIQ brokering group. Select the Origin identity providers from the drop-down list.

Allowed SP: The Allowed SPs are the selected SPs of the trusted providers. The drop-down list displays all the service providers created for the specific brokering group. Select the service providers from the drop-down list

Role: Specify the role you want to validate for the selected Origin identity trusted providers and allowed SP. Click the Validate Rule.

A list is displayed according to the rule validation for the selected trusted providers, role, and permission.



Name: Displays the role name of the selected trusted providers.

Identity Providers: Displays the identity provider name.

Service Providers: Displays the service provider name.

Priority: In ascending order, displays the priority number of the rule validation of the selected trusted providers.

Action: Displays the permission action for validation of the selected trusted providers rule validation.

Role Conditions: Displays the role conditions for the selected trusted providers rule validation. Denial takes precedence over Permit.

Evaluate State: Displays the role conditions evaluate state for the selected trusted providers rule validation. You can see diffferent evaluation states in the role conditions.

Pass 1: If the rule matches the Origin identity provider, allowed service provider or any roles mentioned.

Pass2: If the rule matches the Origin identity provider, allowed service provider or any specific role mentioned.

Ignored: If the rule does not match either Pass 1 or Pass 2 .

Not Executed: The default state of all the roles.

NOTE: If the rule has the evaluate State as Pass 1 action as Deny, then the remaining rules are in the non-executed state.

After a rule has the evaluate state as Pass 2, regardless of the action, the remaining rules are in the non-executed state.

The rules before Pass 1, must have the evaluate state of Ignored. All these ignored rules must have the role condition as **Any**, without specifying any role condition.

Pass 1 evaluation stops, as soon as a match for the Origin identity provider and allowed service provider is found with specific to some role condition.

---

4 Click **Cancel** to close the Rule Validation page.

Generating the Brokering URLs by Using an ID and Target in the Intersite Transfer Service

You can generate the brokering URL's using the ID of the target. You can use this value to simplify the Intersite Transfer Service URL that must be configured at the service provider. For more information, see "Configuring an Intersite Transfer Service Target for a Service Provider" on page 169.

1 In the Administration Console, click **Devices > Identity Servers > Brokering** or click **Devices > Identity Servers > Edit > SAML 2.0 > Trusted Providers > > (Broker Identity under the Service Providers list)** >**Intersite Transfer Service**.

2 **ID:** Specify the ID value of the target.

3 **Target:** Specify the URL of the page that you want to display to users when they authenticate with an Intersite Transfer URL.The behavior of this option is influenced by the **Allow any target** option. If you are using the target ID as part of the Intersite Transfer URL and did not specify a target in the URL, you need to specify the target in this field. For example, if you enter the target URL as it appears below, then it will be displayed when you select **Allow Any Target** option.

```
https://login.company1.com:8443/nidp/saml2/
idpsend?id=217ID&TARGET=https%3A%2F%2FSPBROKER1.labs.blr.novell.com%3A8443%2Fn
idp%2Fsaml2%2Fidpsend%3FPID%3Dhttps%3A%2F%2Flogin.partner2B.com%3A8443%2Fnidp%
2Fsaml2%2Fmetadata%26TARGET%3Dhttps%3A%2F%2Fpartner2b.com
```

4 **Allow any Target:** Select this option to use the target that was specified in the Intersite Transfer URL. If this option is not selected, the target value in the Intersite Transfer URL is ignored and you can see the URL specified in the **Target** option.

Transient Federation within SAML 2.0

You have to make the following configuration changes for the transient federations to work from Origin Identity Provider to SP Broker to Target Service Provider.For example, if the Origin Identity Provider is on SAML 1.0 (transient), the SP Broker and the Target Service Provider also have to be on transient federation.

Origin Identity Provider Configuration

1 Go to **Edit > SAML2 > Trusted Providers > (Broker IDP under the Service Providers list) > Authentication Response**

2 Enable the **Transient Name ID Format** and make it as Default.

Broker Identity Provider Configuration

1 Go to **Edit** > **SAML2** > **Trusted Providers** > **(Origin IDP under the Identity Providers list)** > **Authentication Card** > **Authentication Request**.

2 Select the Transient Name ID Format.

3 Go to **Edit** > **SAML2** > **Trusted Providers** > **(Next hop SP under the Service Providers list)** > **Authentication Response**.

4 Enable the Transient Name ID Format and make it as Default.

Service Provider Configuration

1 Go to **Edit**> **SAML2**> **Trusted Providers** > **(Broker IDP under the Identity Providers list)** > **Authentication Card** > **Authentication Request**.

2 Select the **Transient Name ID Format**

Assigning the Roles for the Origin IDP users in SP Broker Using the Transient Federation Attributes

You can assign the roles for the origin Identity Provider users in Service Provider Brokering using the attributes of the transient federation. When you login as a transient user the federation is authenticated based on roles.

Origin Identity Provider Attribute Configuration

1 In the Administration Console, click **Devices > Identity Servers > Brokering** or click **Devices > Identity Servers > Edit > SAML 2.0 > Trusted Providers > (Broker Identity under the Identity Providers list) > Configuration > Attributes.**

2 Select the Attribute set from the drop-down list.

3 Select the attribute names in the **Available List** and move to **Send with Authentication** list using the arrows.

4 Click **Apply** to map and set the attribute changes to the selected role of the origin identity provider.

Target Service Provider Attribute Configuration

1 In the Administration Console, click **Devices > Identity Servers > Brokering** or click **Devices > Identity Servers > Edit > SAML 2.0 > Service Providers > (Broker Identity under the Service Providers list) > Configuration > Attributes.**

2 Select the Attribute set from the drop-down list.

3 Select the attribute names in the **Available List** and move to **Send with Authentication** list using the arrows.

4 Click **Apply** to map and set the attribute changes to the selected role of the target service provider

Brokering Service Provider Attribute Configuration

The attributes configured in origin identity provider and the target service provider displays the attributes based on the role selected in the brokering service provider attribute configuration available list.

1 In the Administration Console, click **Devices > Identity Servers > Brokering** or click **Devices > Identity Servers > Edit > SAML 2.0 > Service Providers > (Broker Identity under the Service Providers list) > Configuration > Attributes.**

2 Select the Attribute set from the drop-down list.

3 Select the attribute names in the **Available List** and move to **Send with Authentication** list using the arrows.

4 Click **Apply** to map and set the attribute changes to the selected role of the brokering service provider.

Assigning The Local Roles Based On Remote Roles And Attributes

You are able to configure the attributes based on the roles you select in the Attribute set field. You are able to log in and authenticated based on roles federated in the Origin Identity Provider, Target Service Provider and the Brokering Service Provider configuration.

Origin Identity Provider Role Attribute Configuration

1 In the Administration Console, click **Devices > Identity Servers > Shared Settings** >**Attribute Sets > Mapping >New.** The **Add Attribute Mapping** window displays.

2 Select the local attribute name from the drop-down list

3 Enter the remote attribute name for the selected local attribute.

4 Click **OK** to add the remote attribute name. The newly added attribute displays in the Mapping list.

5 In the Administration Console, click **Devices > Identity Servers > Edit > SAML 2.0 > Trusted Providers > (Broker Identity under the Identity Providers list) > Configuration > Attributes.**

6 Select the role from drop-down list in the **Attribute set**.

7 Using the arrows map the attributes in the **Send with Authentication** and **Available List**.

8 Click **Apply** to map the set role and attribute of the origin Identity Provider.

Allowed Service Provider Role Attribute Configuration

1 In the Administration Console, click **Devices > Identity Servers > Shared Settings** >**Attribute Sets > Mapping >New.** The **Add Attribute Mapping** window displays.

2 Select the local attribute name from the drop-down list.

3 Specify the remote attribute name for the selected local attribute.

4 Click **OK**. The newly added attribute displays in the Mapping list.

5 In the Administration Console, click **Devices > Identity Servers > Edit > SAML 2.0 > Service Providers > (Broker Identity under the Service Providers list) > Configuration > Attributes.**

6 Select the role from **Attribute set**.

7 Using the arrows, map the attributes in the **Send with Authentication** and **Available List**.

8 Click **Apply** to map and set the attribute changes to the selected role of the target Identity Service Provider.

Brokering Service Provider Role Attribute Configuration

The roles set and the attribute configured in origin identity provider and the target service provider is added and mapped in the brokering service provider attribute configuration.

1 In the Administration Console, click **Devices > Identity Servers > Shared Settings** >**Attribute Sets > Mapping >New.** The **Add Attribute Mapping** window displays.

2 Select the local attribute name from the drop-down list

3 Enter the remote attribute name for the selected local attribute.

4 Click **OK** to add the remote attribute name. The newly added attribute displays in the Mapping list.

5 In the Administration Console, click **Devices > Identity Servers > Brokering** or click **Devices > Identity Servers > Edit > SAML 2.0 > Service Providers > (Broker Identity under the Service Providers list) > Configuration > Attributes.**

6 Select the role from drop-down list in **Attribute set**.

7 Using the arrows map the attributes in **Send with Authentication** and **Available List**.

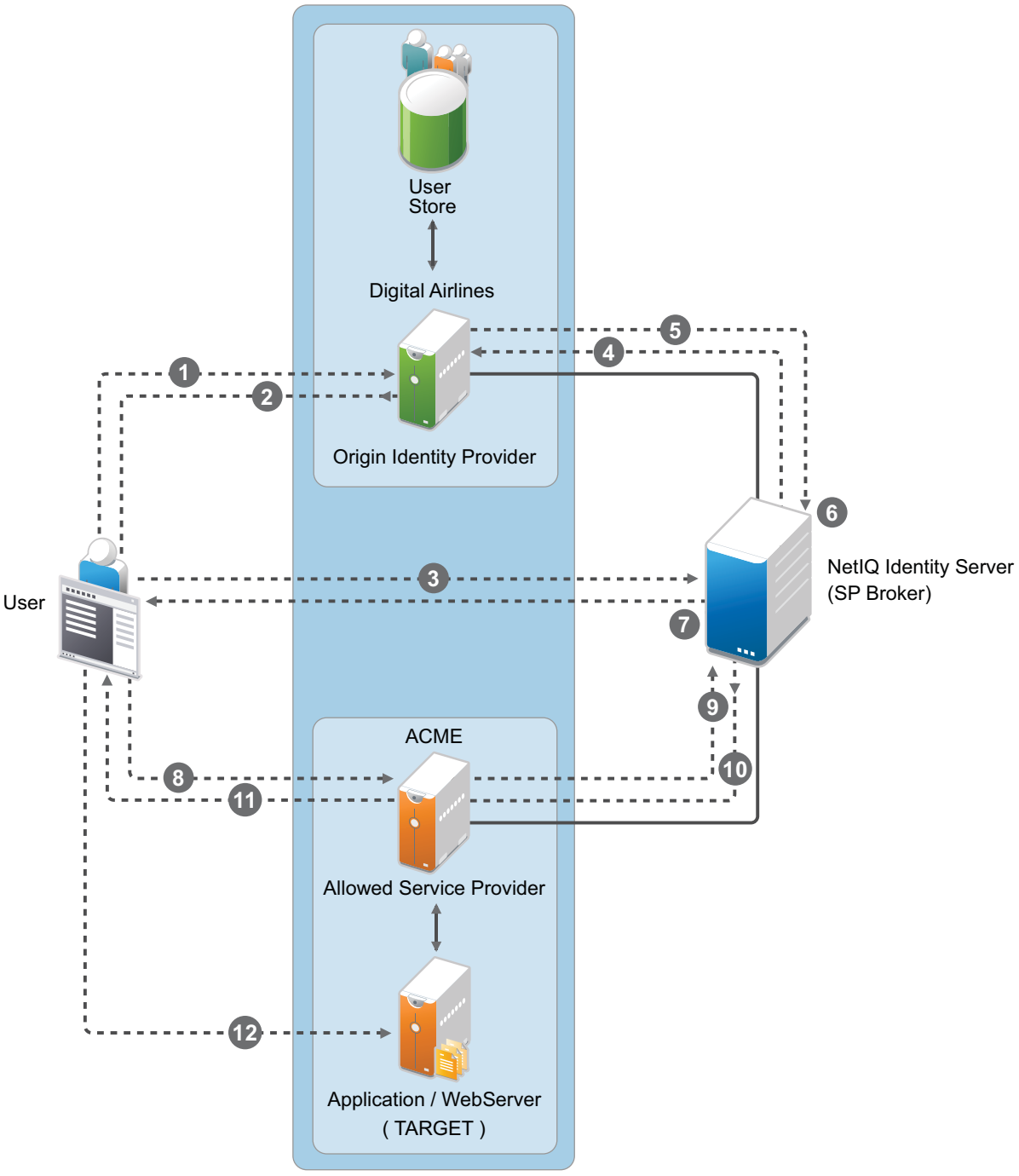
8 Click **Apply** to set the role and configure the attribute mappings.

SP Brokering Example

This example explains how SP Brokering works. Let us assume that two companies Digital Airlines and ACME are business partners. There are certain applications that users of both Digital Airlines and ACME require to access.

With SP Brokering, users in Digital Airlines are provided with an intersite transfer URL that allows users to authenticate at Digital Airlines, set the assertion at ACME, and give access to the target application. With this approach, users do not have to choose from different authentication cards.

The following diagram depicts the SP Brokering workflow:



Workflow:

1. A user is authenticated at Digital Airlines identity provider. The user clicks Broker URL. Digital Airlines checks if this user is authenticated. If not, it asks for user credentials and authenticates the user.
2. Digital Airlines identity provider processes an intersite URL and creates an assertion for SP Broker (NetIQ Identity Server).
3. SP Broker receives the assertion and validates that this assertion is received from a trusted identity provider.

4. SP Broker checks if the trusted identity provider and the service provider (available in the target URL) belong to the same group. SP Broker denies the request if both do not belong to same group.
5. SP Broker sends a request to Digital Airlines identity provider to resolve the artifact.
6. SP Broker receives the SAML assertion from Digital Airlines identity provider and caches attributes/roles received. SP Broker applies any Role policies that have been enabled.
7. SP Broker performs intersite transfer. In the processing of intersite transfer, SP Broker checks if this user was a result of SP Brokering (step 4 earlier). SP Broker enforces the SP Brokering rules check: if any of the rules result in deny, an error page is displayed.
8. SP Broker creates an assertion for ACME.
9. ACME sends a request to SP Broker to resolve the artifact.
10. ACME receives the SAML assertion from the SP Broker along with roles/attributes.
11. ACME sends a redirect to the final target URL. (Note: Redirect happens from ACME's ESP to ACME's identity provider where the user is already authenticated.)
12. The user accesses the target application.

Configuring User Identification Methods for Federation

Configuring authentication involves determining how the service provider interacts with the identity provider during user authentication and federation. Three methods exist for you to identify users from a trusted identity provider:

- You can identify users by matching their authentication credentials
- You can match selected attributes and then prompt for a password to verify the match, or you can use just the attributes for the match.
- You can assume that the user does not have an account and create new accounts with user provisioning. You can also allow for provisioning when the matching methods fail. If there are problems during provisioning, you see error messages with more information.

The following sections describe how to configure these methods:

- "Defining User Identification for Liberty and SAML 2.0" on page 416
- "Defining User Identification for SAML 1.1" on page 419
- "Defining the User Provisioning Method" on page 420
- "User Provisioning Error Messages" on page 422

Defining User Identification for Liberty and SAML 2.0

- "Selecting a User Identification Method for Liberty or SAML 2.0" on page 417
- "Configuring the Attribute Matching Method for Liberty or SAML 2.0" on page 418

Selecting a User Identification Method for Liberty or SAML 2.0

User identification determines how an account at the identity provider is matched with an account at the service provider. If federation is enabled between the two, the user can set up a permanent relationship between the two accounts. If federation is not enabled (see "Configuring a SAML 2.0 Authentication Request" on page 433 and "Configuring a Liberty Authentication Request" on page 464), you cannot set up a user identification method.

1. In the Administration Console, click **Devices > Identity Servers > Edit > Liberty [or SAML 2.0] > [Identity Provider]** > **User Identification**.
2. Specify how users are identified on the SAML 2.0 or Liberty provider. Select one of the following methods:
   - **Authenticate:** Select this option when you want to use login credentials. This option prompts the user to log in at both the identity provider and the service provider on first access. If the user selects to federate, the user is prompted, on subsequent logins, to authenticate only to the identity provider.
     - **Allow 'Provisioning':** Select this option to allow users to create an account when they have no account on the service provider.

       This option requires that you specify a user provisioning method.
   - **Provision account:** Select this option when the users on the identity provider do not have accounts on the service provider. This option allows the service provider to trust any user that has authenticated to the trusted identity provider

     This option requires that you specify a user provisioning method.
   - **Attribute matching:** Select this option when you want to use attributes to match an identity server account with a service provider account. This option requires that you specify a user matching method.
     - **Prompt for password on successful match:** Select this option to prompt the user for a password when the user's name is matched to an account, to ensure that the account matches.
3. Select one of the following:
   - If you selected the **Attribute matching** option, select a method, then click **OK**. If you have not created a matching method, continue with "Configuring the Attribute Matching Method for Liberty or SAML 2.0" on page 418.
   - If you selected the **Provision account** option, select a method, then click **OK**. If you have not created a provisioning method, continue with "Defining the User Provisioning Method" on page 420.
   - If you selected the **Authenticate** option with the **Allow Provisioning** option, select a method, then click **OK**. If you have not created a provisioning method, continue with "Defining the User Provisioning Method" on page 420.
   - If you selected the **Authenticate** option without the **Allow Provisioning** option, click **OK**.
4. Configure the post authentication method.

   Selected Methods: Using the arrow keys to move methods from the **Available Methods** list to the Selected **Methods** list. The selected method is executed when post remote authentication completes.

   For example if you select the passwordfetch method, this method is executed at the service provider after the identity provider authentication and federation completes.

   Logout on method execution failure: If you select this check box, then whenever there is a session failure, the user is logged out automatically.
5. Configure the session options.

Allow IDP to set session timeout: Select Allow Identity Provider to set session timeout between the principal identified by the subject and the SAML authority based on **SessionNotOnOrAfter** attribute in SAML assertion of **authnStatement**.

Overwrite Temporary User: If you select this check box, then the temporary user credentials profile got form previous authentication method in the same session will be overwritten with real user credentials profile got from this authentication method.

Overwrite Real User: If you select this check box, then the real user credentials profile got form previous authentication method in the same session will be overwritten with real user credentials profile got from this authentication method

Assertion Validity Window: You can manually set the assertion validity time for SAML Service Provider (SP) to accommodate clock skew between Service Provider and SAML Identity (IDP) Server.

6 Click **OK** twice, then update the Identity Server.

Configuring the Attribute Matching Method for Liberty or SAML 2.0

If you enabled the **Attribute matching** option when selecting a user identification method, you must configure a matching method.

The Liberty Personal Profile is enabled by default. If you have disabled it, you need to enable it. See "Managing Web Services and Profiles" on page 469.

1 In the Administration Console, click **Devices > Identity Servers > Servers > Edit > Liberty [or SAML 2.0] > [Identity Provider] > User Identification**.

2 Click **Attribute Matching settings**.

3 Select and arrange the user stores you want to use.

Order is important. The user store at the top of the list is searched first. If a match is found, the other user stores are not searched.

4 Select a matching expression, or click **New** to create a look-up expression. For information about creating a look-up expression, see "Configuring User Matching Expressions" on page 63.

5 Specify what action to take if no match is found.

- **Do nothing:** Specifies that an identity provider account is not matched with a service provider account. This option allows the user to authenticate the session without identifying a user account on the service provider.

  IMPORTANT: Do not select this option if the expected name format identifier is persistent. A persistent name format identifier requires that the user be identified so that information can be stored with that user. To support the **Do nothing** option and allow anonymous access, the authentication response must be configured for a transient identifier format. To view the service provider configuration, see "Configuring an Authentication Response for a Service Provider" on page 162.

- **Prompt user for authentication:** Allows the user to specify the credentials for a user that exists on the service provider. Sometimes users have accounts at both the identity provider and the service provider, but the accounts were created independently, use different names (for example, joe.smith and jsmith) and different passwords, and share no common attributes except for the credentials known by the user.

- **Provision account:** Assumes that the user does not have an account at the service provider and creates one for the user. You must create a provisioning method.

6 Click **OK**.

7 (Conditional) If you selected **Provision account** when no match is found, select the **Provision settings** icon. For information about this process, see "Defining the User Provisioning Method" on page 420.

8 Click **OK** twice, then update the Identity Server.

Defining User Identification for SAML 1.1

- "Selecting a User Identification Method for SAML 1.1" on page 419
- "Configuring the Attribute Matching Method for SAML 1.1" on page 420

Selecting a User Identification Method for SAML 1.1

Two methods exist for identifying users from an identity provider when using the SAML 1.1 protocol. You can specify that no account matching needs to occur, or you can configure a match method. You configure a match method when you want to use attributes from the identity provider to uniquely identify a user on the service provider.

1 In the Administration Console, click **Devices > Identity Servers > Edit > SAML 1.1 > [Identity Provider]** > **User Identification**.

2 In the **Satisfies contract** option, specify the contract that can be used to satisfy the assertion received from the identity provider. Because SAML 1.1 does not use contracts and because the Identity Server is contract-based, this setting permits an association to be made between a contract and a SAML 1.1 assertion.

   Use caution when assigning the contract to associate with the assertion, because it is possible to imply that authentication has occurred, when it has not. For example, if a contract is assigned to the assertion, and the contract has two authentication methods (such as one for name/password and another for X.509), the server sending the assertion might use only name/password, but the service provider might assume that X.509 took place and then incorrectly assert it to another server.

3 Select one of the following options for user identification:

   - **Do nothing:** Specifies that an identity provider account is not matched with a service provider account. This option allows the user to authenticate the session without identifying a user account on the service provider.
   - **Attribute matching:** Authenticates a user by matching a user account on the identity provider with an account on the service provider. This option requires that you set up the match method.
     - **Prompt for password on successful match:** Specifies whether to prompt the user for a password when the user is matched to an account, to ensure that the account matches.

4 Select one of the following:

   - If you selected **Do nothing**, continue with Step 6.
   - If you selected **Attribute matching**, continue with "Configuring the Attribute Matching Method for SAML 1.1" on page 420.

5 You can also configure the assertion time manually.

   - **Assertion Validity Window:** You can manually set the assertion validity time for SAML Service Provider (SP) to accommodate clock skew between Service Provider and SAML Identity (IDP) Server.

6 Click **OK** twice.

7 Click **Apply** to make the user identification configuration changes.

8 Update the Identity Server.

Configuring the Attribute Matching Method for SAML 1.1

A user matching expression is a set of logic groups with attributes that uniquely identify a user. User matching expressions enable you to map the Liberty attributes to the correct LDAP attributes during searches. You must know the LDAP attributes that can be used to identify unique users in the user store.

To use user matching, the Personal Profile must be enabled. It is enabled by default. If you have disabled it, you need to enable it. See "Managing Web Services and Profiles" on page 469.

1 In the Administration Console, click **Devices > Identity Servers > Servers > Edit > SAML 1.1 > [Identity Provider] > User Identification.**

2 To configure the match method, click **Attribute Matching settings**.

3 Select and arrange the user stores you want to use.

Order is important. The user store at the top of the list is searched first. If a match is found, the other user stores are not searched.

4 Select a matching expression, or click **New** to create a look-up expression. For information about creating a look-up expression, see "Configuring User Matching Expressions" on page 63.

5 Click **OK**.

6 Update the Identity Server.

Defining the User Provisioning Method

If you have selected **Provision account** as the user identification method or have created an attribute matching setting that allows for provisioning when no match is found, you need to create a provision method. This procedure involves selecting required and optional attributes that the service provider requests from the identity provider during provisioning.

IMPORTANT: When a user object is created in the directory, some attributes are initially created with the value of NAM Generated. Afterwards, an attempt is made to write the required and optional attributes to the new user object. Because required and optional attributes are profile attributes, the system checks the write policy for the profile's Data Location Settings (specified in **Liberty** > **Web Service Provider**) and writes the attribute in either LDAP or the configuration store. In order for the LDAP write to succeed, each attribute must be properly mapped as an LDAP Attribute. Additionally, you must enable the read/write permissions for each attribute in the Liberty/LDAP attribute maps. See "Mapping LDAP and Liberty Attributes" on page 477.

To configure user provisioning:

1 In the Administration Console, click **Devices > Identity Servers > Servers > Edit > Liberty [or SAML 2.0] > [Identity Provider]** > **User Identification**.

2 Click the **Provisioning settings** icon.

3 Select the required attributes from the **Available Attributes** list and move them to the **Attributes** list.

Required attributes are those used in the creation of a user name, or that are required when creating the account.

4 Click **Next**.

5 Select optional attributes from the **Available Attributes** list and move them to the **Attributes** list.

This step is similar to selecting required attributes. However, the user provisioning request creates the user account whether or not the optional attributes exist on the service provider.

6 Click **Next**.

7 Define how to create the username.

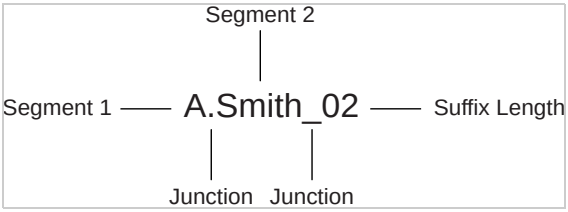
You can specify whether users are prompted to create their own usernames or whether the system automatically creates usernames. Selecting an attribute for the username segments from the required attributes list improves the chances that a new username is successfully created.

Maximum length: The maximum length of the user name. This value must be between 1 and 50.

Prompt for user name: Enables users to create their own usernames.

Automatically create user name: Specifies that the system creates usernames. You can configure the segments for the system to use when creating usernames and configure how the names are displayed.

For example, if you are using the required attributes of Common First Name and Common Last Name, a username for Adam Smith might be generated as A.Smith_02, as shown in the following illustration:



Use the following settings to specify how this is accomplished:

- **Segment 1:** The required attribute to use as the first segment for the user name. The values displayed in this drop-down menu correspond to the required attributes you selected. For example, you might select Common First Name to use for **Segment 1**.
- **Length:** The length of the first attribute segment. For example, if you selected Common First Name for the **Segment 1** value, setting the length to 1 specifies that the system uses the first letter of the Common First Name attribute. Therefore, Adam Smith would be ASmith.
- **Junction:** The type of junction to use between the attributes of the user name. If a period is selected, Adam Smith would display as A.Smith.
- **Segment 2:** The required attribute to use as the second segment for the user name. The values displayed in this drop-down menu correspond to the required attributes you selected. For example, you might select Common Last Name to use for **Segment 2**.
- **Length:** The length of the second attribute segment. For example, if you selected Common Last Name for the **Segment 2** value, you might set the length to **All**, so that the full last name is displayed. However, the system does not allow more than 20 characters for the length of segment 2.
- **Ensure name is unique:** Applies a suffix to the colliding name until a unique name is found, if using attributes causes a collision with an existing name. If no attributes are provided, or the lengths for them are 0, and this option is selected, the system creates a unique name.

8 Click **Next**.

9 Specify password settings.

Use this page to specify whether to prompt the user for a password or to create a password automatically.

Min. password length: The minimum length of the password.

Max. password length: The maximum length of the password.

Prompt for password: Prompts the user for a password.

Automatically create password: Specifies whether to automatically create passwords.

10 Click **Next**.

11 Specify the user store and context in which to create the account.

User Store: The user store in which to create the new user account.

Context: The context in the user store you want accounts created.

The system creates the user within a specific context; however, uniqueness is not guaranteed across the directory.

Delete user provisioning accounts if federation is terminated: Specifies whether to automatically delete the provisioned user account at the service provider if the user terminates his or her federation between the identity provider and service provider.

12 Click **Finish**.

13 Click **OK** twice, then update the Identity Server.

User Provisioning Error Messages

The following error messages are displayed for the end user if there are problems during provisioning:

Table 5-14 *Provisioning Error Messages*

Error Message	Cause
`Username length cannot exceed (?) characters.`	The user entered more characters for a user name than is allowed, as specified by the administrator.
`Username is not available.`	The user entered a name that already exists in the directory.
`Passwords don't match.`	The user provided two password values that do not match.
`Passwords must be between (x) and (y) characters in length.`	The user provided password values that are either too short or too long.
`Username unavailable.`	The provisioned user account was deleted without first defederating the user. Remove orphaned identity objects from the configuration datastore. **IMPORTANT:** Only experienced LDAP users must remove orphaned identity objects from the configuration datastore. You must ensure that the objects you are removing are orphaned. Otherwise, you create orphaned objects by mistake.

Error Message	Cause
`Unable to complete authentication request.`	The password provided does not conform to the Windows password complexity policy in Active Directory. Ensure that Active Directory is configured to use a secure port, such as 636, and that the user's password conforms to the complexity policy. If you encounter this error, you must reset the password on the Windows machine. Can occur when users are allowed to create accounts from a service provider's login page, when the service provider uses Active Directory for the user store.

Configuring SAML 2.0

This section explains how to use the SAML 2.0 protocol to set up the trust with internal and external identity providers, service providers, and Embedded Service Providers (ESPs). Topics include:

- "Understanding How Access Manager Uses SAML" on page 423
- "Configuring a SAML 2.0 Profile" on page 427
- "Creating a SAML 2.0 Service Provider" on page 428
- "Creating a SAML 2.0 Identity Provider" on page 428
- "Executing Authorization Based Roles Policy During SAML 2.0 Service Provider Initiated Request" on page 429
- "Contracts Assigned to a SAML 2.0 Service Provider" on page 430
- "Editing a SAML 2.0 Service Provider's Metadata" on page 433
- "Configuring a SAML 2.0 Authentication Request" on page 433
- "Configuring the SAML 2.0 Authentication Response" on page 437
- "Defining Options for SAML 2.0" on page 438
- "Defining Session Synchronization for the A-Select SAML 2.0 Identity Provider" on page 443
- "Configuring the Liberty or SAML 2.0 Session Timeout" on page 443
- "Modifying the Authentication Card for Liberty or SAML 2.0" on page 444
- "Configuring Multiple SAML 2.0 Service Providers on the Same Host for a Single SAML Identity Provider" on page 445
- "Configuring Active Directory Federation Services with SAML 2.0 for Single Sign-On" on page 445

Understanding How Access Manager Uses SAML

Security Assertions Markup Language (SAML) is an XML-based framework for communicating security assertions (user authentication, entitlement, and attribute information) between trusted identity providers and trusted service providers. For example, an airline company can make assertions to authenticate a user to a partner company or another enterprise application, such as a car rental company or hotel.

Identity Server allows SAML assertions to be exchanged with trusted service providers that use SAML servers. Using SAML assertions in each Access Manager component protects confidential information by removing the need to pass user credentials between the components to handle session management.

An identity provider using the SAML protocol generates and receives assertions for authentication, according to the SAML 1.0, 1.1, and 2.0 specifications described on the Oasis Standards Web site (http://www.oasis-open.org/specs/index.php).

This section describes how Access Manager uses SAML. It includes the following topics:

- "Attribute Mapping with Liberty" on page 424
- "Trusted Provider Reference Metadata" on page 424
- "Identity Provider Process Flow" on page 425
- "SAML Service Provider Process Flow" on page 426

Attribute Mapping with Liberty

Attribute-based mapping involves one website communicating identity information about a subject to another website to support transactions. However, the identity information might be some characteristic of the subject, such as a role. The attribute-based mapping is important when the subject's identity is either not important, must not be shared, or is insufficient on its own.

To interoperate with trusted service providers through the SAML protocol, Identity Server distinguishes between different attributes from different SAML implementations. Access Manager uses Liberty attributes in the SAML administration. When you specify which attributes to include in an assertion, or which attributes to use when locating the user from an assertion, these attributes must always be specified in the Liberty format.

In an attribute map, SAML attributes from each vendor's implementation is converted to Liberty attributes. (See "Configuring Attribute Sets" on page 61.)

You can find detailed information about SAML 2.0 on the OASIS Standards Web site (http://www.oasis-open.org/specs/).

Trusted Provider Reference Metadata

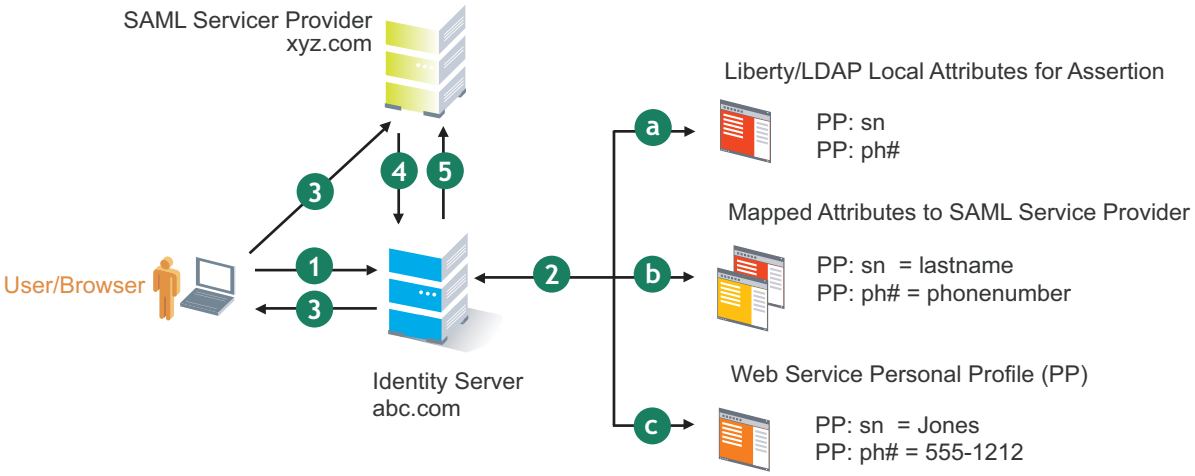
Identity Server generates metadata for server communication and identification. You can obtain metadata through URL or XML document and then enter it in the system while creating the reference. Metadata is traded with federation partners and supplies various information regarding contact and organization information located at Identity Server. Metadata is generated automatically for SAML 2.0. You enter it manually for SAML 1.1.

IMPORTANT: The SAML 2.0 and Liberty 1.2 protocols define a logout mechanism whereby the service provider sends a logout command to the trusted identity provider when a user logs out at a service provider. SAML 1.1 does not provide such a mechanism. For this reason, when a logout occurs at the SAML 1.1 service provider, no logout occurs at the trusted identity provider. A valid session is still running at the identity provider, and no credentials need to be entered. To log out at both providers, users must navigate to the identity provider that authenticated them to the SAML 1.1 service provider and log out manually.

Identity Provider Process Flow

The following illustration provides an example of an Identity Server automatically creating an authenticated session for the user at a trusted SAML service provider. PP indicates a Personal Profile Service as defined by the Liberty specification.

Figure 5-20 *SAML Service Provider Process Flow*



1. A user is logged in to Identity Server at abc.com (the user's identity provider) and clicks a link to xyz.com, a trusted SAML service provider.

   Identity Server at abc.com generates the artifact. This starts the process of generating and sending the SAML assertion. The HREF would look similar to the following:

   ```
   http://nidp.com/saml/genafct?TARGET=http://xyz.com/index.html&AID=XYZ
   ```

2. Identity Server processes attributes as follows:
   a. The server looks up for LDAP or Liberty-LDAP mapped attributes. (See "Mapping LDAP and Liberty Attributes" on page 477.) In this example, you use Liberty attributes such as *PP: sn* instead of *surname*. *PP: sn* and *PP: ph#* are attributes that you are sending to xyz.com.
   b. Identity Server processes these attributes with a SAML implementation-specific attribute.

      Because the identity provider must interoperate with other SAML service providers that probably do not use consistent attribute names, you can map the service provider attributes to your Liberty and LDAP attributes on Identity Server. In this example, the service provider names for the Liberty *PP: sn* and *PP: ph#* attributes are *lastname* and *phonenumber*, respectively. (See "Configuring the Attributes Obtained at Authentication" on page 156.)
   c. Identity Server uses the PP service to look up the values for the user's *PP: sn* and *PP: ph#* attributes.

      Identity Server recognizes that the values for the user's *PP: sn* and *PP: ph#* attributes are *Jones* and *555-1212*, respectively.
3. Identity Server sends an HTTP redirect with an artifact.

   Identity Server now has the information to generate a SAML assertion. Identity Server sends an HTTP redirect containing the artifact to the browser. The redirect looks similar to the following:

   ```
   http://xyz.com/auth/afct?TARGET=http://xyz.com/index.html&SAMLArtifact
   =<<artifact>>
   ```

4. The remote SAML server requests the assertion.

   The HTTP redirect results in the browser sending the artifact to the SAML server at xyz.com. The SAML server at xyz.com requests the SAML assertion from Identity Server.
5. Identity Server sends the assertion to the remote SAML server.

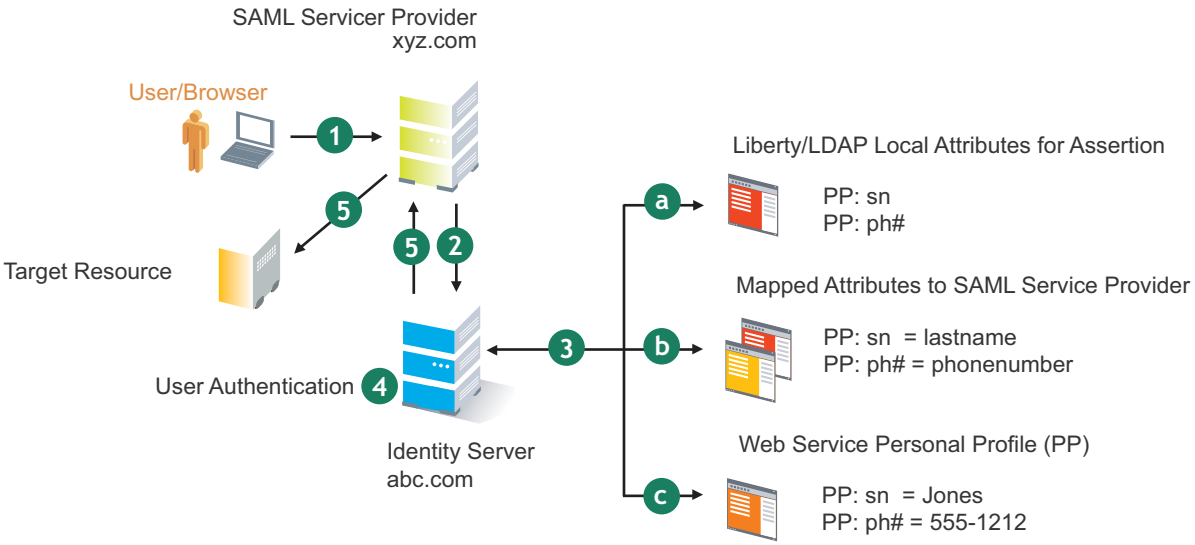
The remote SAML server receives the artifact and looks up the assertion.The assertion is sent to the SAML server at *xyz.com* in a SOAP envelope. The assertion contains the attributes *lastname=Jones* and *phonenumber=555-1212*.

The user now has an authenticated session at xyz.com. The xyz.com SAML server redirects the user's browser to http://xyz.com/index.html, which was referenced in the original HREF in Step 1.

SAML Service Provider Process Flow

The following illustration provides an example of the authentication process on the consumer side, when a user clicks a link at the SAML service provider (xyz.com) to begin an authentication session with an identity provider (such as abc.com). PP indicates a Personal Profile Service as defined by the Liberty specification.

Figure 5-21 *SAML Consumer Process Flow*



1. The user clicks a link at xyz.com.

   This generates a SAML assertion intended for Identity Server at abc.com, which is the identity provider in an Access Manager configuration. After the SAML server generates the artifact, it sends the browser a redirect containing the artifact. The browser is redirected to the identity provider, which receives the artifact. The URL sent to Identity Server looks similar to the following:

   ```
   http://nidp.com/auth/afct?TARGET=http://abc.com/index.html&SAMLArtifact
   =<<artifact>>
   ```

2. Identity Server at abc.com receives the assertion.

   The assertion is sent to Identity Server packaged in a SOAP envelope. In this example, the assertion contains the attributes *lastname=Jones*, and *phonenumber=555-1212*.

3. Identity Server determines which attributes to use when locating the user.

   Identity Server must determine how to locate the user in the directory. When you created the SAML service provider reference for xyz.com, you specified which Liberty attributes must be used for this purpose. In this case, the you specified that *PP: sn* and *PP: ph#* must be used.

   a. Identity Server processes the Liberty attribute map (see "Mapping LDAP and Liberty Attributes" on page 477) to the SAML implementation-specific attributes (see "Configuring the Attributes Obtained at Authentication" on page 156).

Because this SAML implementation must interoperate with other SAML implementations that probably do not use consistent attribute names, you can map the attributes used by each third-party SAML implementation to Liberty attributes on Identity Server.

b. Identity Server receives implementation-specific SAML attribute names.

   The trusted service provider's names for the Liberty *PP: sn* and *PP: ph#* attributes are returned. Using the attribute map, Identity Server knows that the service provider's names for these attributes are *lastname* and *phonenumber*, respectively.

c. Identity Server uses the PP service to lookup the values for the user's *PP: sn* and *PP: ph#* attributes.

   Identity Server now recognizes that the values for the user's *PP: sn* and *PP: ph#* attributes are *Jones* and *555-1212*, respectively. The user's DN is returned to Identity Server, and the user is authenticated.

4. The user's DN is returned to Identity Server, and the user is authenticated.
5. The user is redirected to the target resource at xyz.com.

Configuring a SAML 2.0 Profile

You can configure the methods of communication that are available at the server for requests and responses sent between providers. These settings affect the metadata for the server and must be determined prior to publishing to other sites.

Profiles control the methods of communication that are available for SAML 2.0 protocol requests and responses sent between trusted providers. The identity provider uses the incoming metadata to determine how to respond.

All available profile bindings are enabled by default. SOAP is used when all are enabled (or if the service provider has not specified a preference), followed by HTTP Post, then HTTP Redirect.

1 In the Administration Console, click **Devices > Identity Servers > Edit > SAML 2.0 > Profiles**.

2 Configure the following fields for identity providers and identity consumers (service providers):

   Artifact Resolution: Specify whether to enable artifact resolution for the identity provider and identity consumer.

   The assertion consumer service at the service provider performs a back-channel exchange with the artifact resolution service at the identity provider. Artifacts are small data objects pointing to larger SAML protocol messages. They are designed to be embedded in URLs and conveyed in HTTP messages.

   Login: Specifies the communication channel to use when the user logs in. Select one or more of the following:

   - **Post:** A browser-based method used when the SAML requester and responder need to communicate through an HTTP user agent. This occurs when the communicating parties do not share a direct path of communication. You also use this when the responder requires user interaction to fulfill the request, such as when the user must authenticate to it.
   - **Redirect:** A browser-based method that uses HTTP 302 redirects or HTTP GET requests to communicate requests from this identity site to the service provider. SAML messages are transmitted within URL parameters.

Single Logout: Specifies the communication channel to use when the user logs out. Select one or more of the following:

- **HTTP Post:** A browser-based method used when the SAML requester and responder need to communicate by using an HTTP user agent. This occurs, for example, when the communicating parties do not share a direct path of communication. You also use this when the responder requires user interaction in order to fulfill the request, such as when the user must authenticate to it.
- **HTTP Redirect:** A browser-based method that uses HTTP 302 redirects or HTTP GET requests to communicate requests from this identity site to the service provider. SAML messages are transmitted within URL parameters.
- **SOAP:** Uses SOAP back channel over HTTP messaging to communicate requests from this identity provider to the service provider.

NOTE: If **Show logged out providers** option (**Identity Servers > Edit > Identity Providers**) is enabled with HTTP Post profile, logout request from service provider does not complete. This is due to the difference in the HTTP method used in the logout request. It is recommended to use HTTP Redirect method when **Show logged out providers option** is enabled.

Name Management: Specifies the communication channel for sharing the common identifiers for a user between identity providers and service providers. When an identity provider has exchanged a persistent identifier for the user with a service provider, the providers share the common identifier for a length of time. When either the identity provider or service provider changes the format or value to identify the user, the system can ensure that the new format or value is properly transmitted. Select one or more of the following:

- **HTTP Post:** A browser-based method used when the SAML requester and responder need to communicate using an HTTP user agent. This occurs, for example, when the communicating parties do not share a direct path of communication. You also use this when the responder requires user interaction in order to fulfill the request, such as when the user must authenticate to it.
- **HTTP Redirect:** A browser-based method that uses HTTP 302 redirects or HTTP GET requests to communicate requests from this identity site to the service provider. SAML messages are transmitted within URL parameters.
- **SOAP:** Uses SOAP back channel over HTTP messaging to communicate requests from this identity provider to the service provider.

3 Click **OK**, then update Identity Server.

4 (Conditional) If you have set up trusted providers and have modified these profiles, the providers need to reimport the metadata from this Identity Server.

Creating a SAML 2.0 Service Provider

See "Creating a Trusted Service Provider" on page 152.

Creating a SAML 2.0 Identity Provider

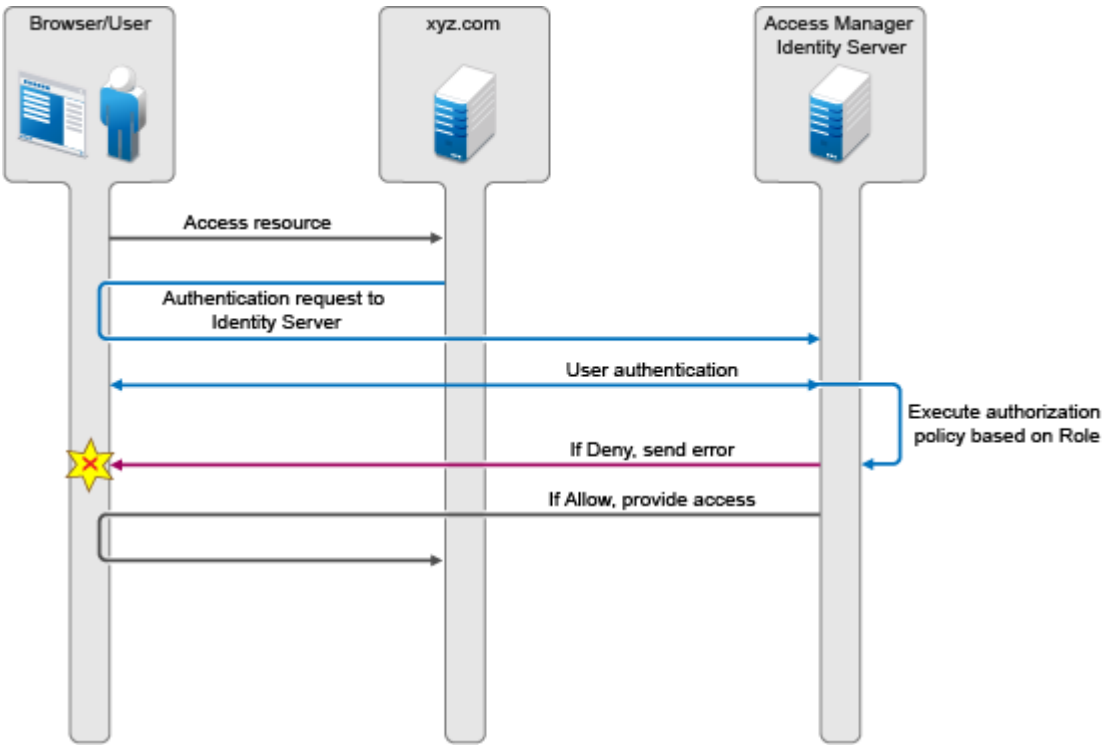
See "Creating a Trusted Identity Provider" on page 150.

Executing Authorization Based Roles Policy During SAML 2.0 Service Provider Initiated Request

Access Manager service provider federation profiles do not currently allow control based on authorization policies. Usually the service providers enforce authorization rules. However, not all service providers have this flexibility. It is recommended not to trust the service provider to enforce such rules. You can now apply an authorization policy to a configured service provider to either allow or not to allow access to the service provider. Identity Server will evaluate the service provider and will generate the successful/unsuccessful assertions.

Use Case: Company xyz.com uses a CRM application that is protected through the SAML 2.0 service provider. This application must only be accessible to the sales team and not to any other teams. Whenever a user accesses the application through the service provider, it redirects to Identity Server for validating the user.

Figure 5-22 *Executing Authorization Policy Based on Role*



Identity Server must authenticate the user and then check if the user is a member of the sales team. If yes, Identity Server sends a successful assertion to the service provider. Else, Identity Server sends an error response to the service provider.

By default, Identity Server executes these authorization policies after a user is authenticated during spsend. Set the `ALLOW_AUTH_POLICY_EXECUTION` option to false to disable Identity Server to execute the authorization policies. For information about how to set this option, see “Configuring Identity Server Global Options” on page 57.

If the authorization policy is to deny execution, Identity Server sends the following message as part of an assertion response. `<samlp:Status> <samlp:StatusCode Value="urn:oasis:names:tc:SAML:2.0:status:Responder"> <samlp:StatusCodeValue="urn:oasis:names:tc:SAML:2.0:status:RequestDenied" /> </ samlp:StatusCode> <StatusMessage>Authorization is failed</StatusMessage> </ samlp:Status>`

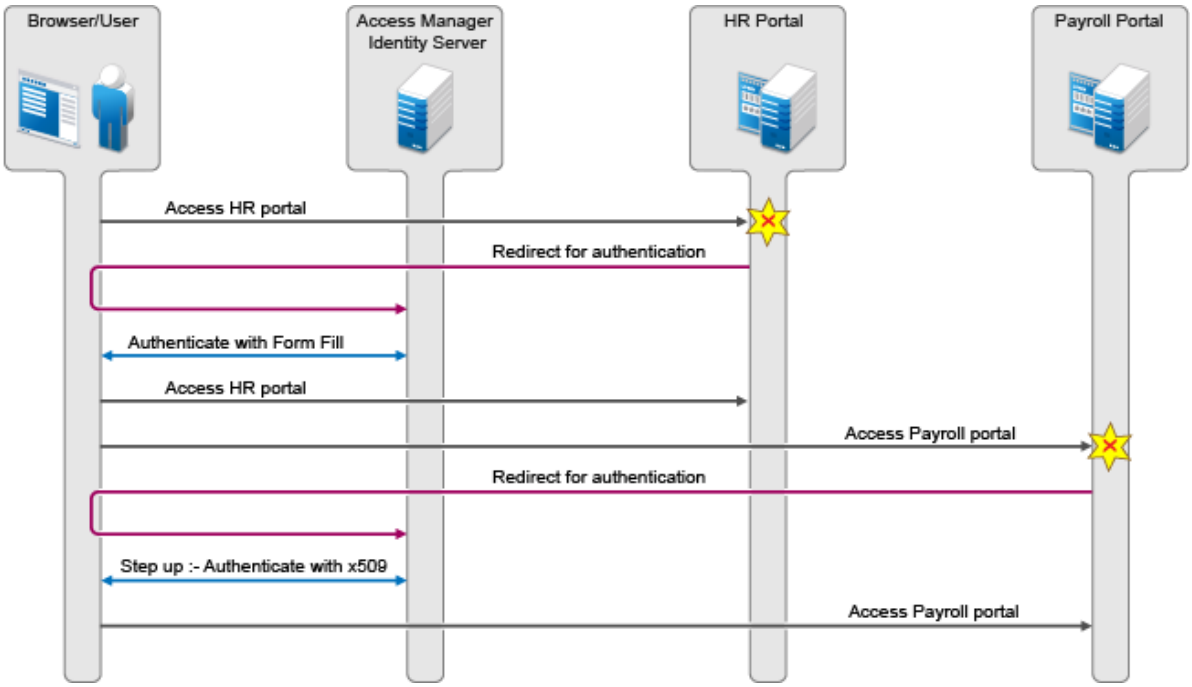
For more information about configuring a brokering for authorization of service providers, see “Configuring a Brokering for Authorization of Service Providers” on page 404.

Contracts Assigned to a SAML 2.0 Service Provider

During federation, when a service provider initiates an authentication request, contract information may not be available. If the contract information is not available, Identity Server executes a default contract for validating the user. The step up authentication feature enables you to assign a default contract for service providers in such scenarios.

The following scenario helps you understand the execution of contracts that are assigned to the SAML 2.0 service provider.

Figure 5-23 *Step Up Authentication example with two applications:*



There are two applications Payroll and HR web applications protected through different service providers and are using Access Manager Identity Server as identity provider. The user wants to use the name/password form contract whenever the user accesses the HR application and wants to use the higher level contract say X509 for the Payroll application. Identity Server provides ability to execute the appropriate contract that has been assigned to the service provider instead of executing the default contract.

The following procedure allows you to assign a specific contract to the service provider.

1. Click on **Devices > Identity Servers > Edit > > SAML2.0**.
2. Click on configured service provider.
3. Go to **Options** > **Step Up Authentication** contracts and select the contracts from the **Available contracts** list.

The following table lists the behavior of a service provider request.

Service Provider Request	Result (Identity Server Response if the user is not authenticated)
Service provider request has no contract information to be executed at Identity Server.	
1. Identity Server has no contracts set for this service provider as in Step 3.	Execute default contract for validating the user and default contract name will be sent in the response.
2. Identity Server has contract C1 set for this service provider as in Step 3.	C1 will be executed for validating the user and C1 will be sent in response.
Service provider requests execution of contract C1 at Identity Server.	
1. Identity Server has no contracts set for this service provider as in Step 3.	C1 will be executed for validating the user and C1 will be sent in response.
2. Identity Server has contract C1 set for this service provider as in Step 3.	C1 will be executed for validating the user and C1 will be sent in response.
3. Identity Server has contract C2 set for this service provider. C2 has trust level check disabled.	C2 will be executed for validating the user and C2 will be sent in response. (**Note**: This is as good as C1 not available at Identity Server)
4. Identity Server has contract C2 set for this service provider. C2 has trust level check enabled.	If trust level of C2 >= trust level of C1, then C2 will be executed and C2 will be sent in response. If trust level of C2 < trust level of C1, then C1 will be executed and C1 will be sent in response. If C1 is not available at Identity Server, then C2 is executed and C2 is sent in the response.

NOTE: When using the service provider (SP) initiated login with a SAML 2.0 SP federation, the SP configuration can impact the selection of the Access Manager contract for authentication depending on the values sent in SAML authentication request. To make it work properly, you must define your Access Manager contract URI to match with the request sent by the service provider.

For more information, see “Allowable Class” in “Configuring Authentication Contracts” on page 289.

Configuring Communication Security for a SAML 2.0 Identity Provider

The security settings control the direct communication between Identity Server and the identity provider across the SOAP back channel.

1. In the Administration Console, click **Devices > Identity Servers > Edit** > **SAML 2.0**.
2. Click the name of an identity provider.
3. On the Trust page, fill in the following fields:

   Name: Specify the display name for this trusted provider. The default name is the name you entered when creating the trusted provider.

   The **Security** section specifies how to validate messages received from trusted providers over the SOAP back channel. Both the identity provider and the service provider in the trusted relationship must be configured to use the same security method.

   Encrypt name identifiers: Specifies whether you want the name identifiers encrypted on the wire.

Select one of the following security methods:

- **Message Signing:** Relies upon message signing using a digital signature.
- **Mutual SSL:** Specifies that this trusted provider provides a digital certificate (mutual SSL) when it sends a SOAP message.

  SSL communication requires only the client to trust the server. For mutual SSL, the server must also trust the client. For the client to trust the server, the server's certificate authority (CA) certificate must be imported into the client trust store. For the server to trust the client, the client's CA certificate must be imported into the server trust store.
- **Basic Authentication:** Specifies standard header-based authentication. This method assumes that a name and password for authentication are sent and received over the SOAP back channel.

  Send: The name and password to be sent for authentication to the trusted partner. The partner expects this password for all SOAP back-channel requests, which means that the name and password must be agreed upon.

  Verify: The name and password used to verify data that the trusted provider sends.

  Certificate Revocation Check Periodicity: Specifies if the certificate is valid or not. You can define periodicity to validate on start up, on assertion level, or set frequency to hourly/daily.

4 Click **OK** twice.

5 Update Identity Server.

Configuring Communication Security for a SAML 2.0 Service Provider

The security settings control the direct communication between Identity Server and the service provider across the SOAP back channel.

1 In the Administration Console, click **Devices > Identity Servers > Edit** > **SAML 2.0**.

2 Click the name of a service provider.

3 On the Trust page, fill in the following fields:

Name: Specify the display name for this trusted provider. The default name is the name you entered when creating the trusted provider.

The **Security** section specifies how to validate messages received from trusted providers over the SOAP back channel. Both the identity provider and the service provider in the trusted relationship must be configured to use the same security method.

Encrypt assertions: Specifies whether you want the assertions encrypted on the wire.

Encrypt name identifiers: Specifies whether you want the name identifiers encrypted on the wire.

SOAP Back Channel Security Method: Select one of the following security methods.

- **Message Signing:** Relies upon message signing using a digital signature.
- **Mutual SSL:** Specifies that this trusted provider provides a digital certificate (mutual SSL) when it sends a SOAP message.

  SSL communication requires only the client to trust the server. For mutual SSL, the server must also trust the client. For the client to trust the server, the server's certificate authority (CA) certificate must be imported into the client trust store. For the server to trust the client, the client's CA certificate must be imported into the server trust store.
- **Basic Authentication:** Specifies standard header-based authentication. This method assumes that a name and password for authentication are sent and received over the SOAP back channel.

Send: The name and password to be sent for authentication to the trusted partner. The partner expects this password for all SOAP back-channel requests, which means that the name and password must be agreed upon.

Verify: The name and password used to verify data that the trusted provider sends.

4 Click **OK** twice.

5 Update Identity Server.

Editing a SAML 2.0 Service Provider's Metadata

See "Editing a SAML 2.0 Service Provider's Metadata" on page 159.

Configuring a SAML 2.0 Authentication Request

You can configure how an authentication request is federated. When users authenticate to a service provider, they can be given the option to federate their account identities with the preferred identity provider. This process creates an account association between the identity provider and service provider that enables single sign-on and single log-out.

The authentication request specifies how you want the identity provider to handle the authentication process so that it meets the security needs of Identity Server.

1 In the Administration Console, click **Devices > Identity Servers > Edit > SAML 2.0 > [Identity Provider] > Authentication Card** > **Authentication Request.**

2 Configure the name identifier format:

Persistent: A persistent identifier federates the user profile on the identity provider with the user profile on the service provider. It remains intact between sessions.

The persistent identifier is saved to the user data store and hides the user's identity to prevent tracking of user activities across different relying parties.

- **After authentication:** Specifies that the persistent identifier can be sent after the user has authenticated (logged in) to the service provider. When you set only this option, users must log in locally. Because the user is required to authenticate locally, you do not need to set up user identification.
- **During authentication:** Specifies that the persistent identifier can be sent when the user selects the authentication card of the identity provider. Typically, a user is not authenticated at the service provider when this selection is made. When the identity provider sends a response to the service provider, the user needs to be identified on the service provider. If you enable this option, ensure that you configure a user identification method. See "Selecting a User Identification Method for Liberty or SAML 2.0" on page 417.

Transient: Specifies that a transient identifier, which expires between sessions, can be sent.

Unspecified: Allows either a persistent or a transient identifier to be sent.

3 Select one of the following options for the **Requested By** option:

Do not specify: Specifies that the identity provider can send any type of authentication to satisfy a service provider's request, and instructs a service provider to not send a request for a specific authentication type or contract.

Use Types: Specifies that authentication types must be used.

Select the type of comparison (for more information, see "Understanding Comparison Contexts" on page 436):

- **Exact:** Indicates that the class or type specified in the authentication statement must be an exact match to at least one contract.

- **Minimum:** Indicates that the contract must be as strong as the class or type specified in the authentication statement.
- **Better:** Indicates the contract that must be stronger than the class or type specified in the authentication statement.
- **Maximum:** Indicates that contract must as strong as possible without exceeding the strength of at least one of the authentication contexts specified.

Select the types from the **Available types** field to specify which type to use for authentication between trusted service providers and identity providers. Standard types include Name/Password, X.509, Token, and so on.

Use Contracts: Specifies that authentication contracts must be used.

Select the type of comparison (for more information, see "Understanding Comparison Contexts" on page 436):

- **Exact:** Indicates that the class or type specified in the authentication statement must be an exact match to at least one contract.
- **Minimum:** Indicates that the contract must be as strong as the class or type specified in the authentication statement.
- **Better:** Indicates the contract that must be stronger than the class or type specified in the authentication statement.
- **Maximum:** Indicates that contract must as strong as possible without exceeding the strength of at least one of the authentication contexts specified.

Select the contract from the **Available contracts** list. For a contract to appear in the **Available contracts** list, the contract must have the **Satisfiable by External Provider** option enabled. To use the contract for federated authentication, the contract's URI must be the same on the identity provider and the service provider. For information about contract options, see "Configuring Authentication Contracts" on page 289.

Most third-party identity providers do not support contracts.

4 Configure the options:

Response protocol binding: Select **Artifact** or **Post** or **Let IDP Decide**. Artifact and Post are the two methods for transmitting assertions between the authenticating system and the target system.

If you select **Let IDP Decide**, the binding is selected based on the profile that is enabled at Identity Provider and the binding selected in the service provider.

NOTE: The post binding can be configured to be sent as a compressed option. Perform the following steps to achieve this:

1. In the Administration Console, click **Devices > Identity Servers > Edit** > **Options** > **New**.
2. Select **IS SAML2 POST INFLATE** in **Property Type** and **true** in **Property Value**.
3. Click **OK**.
4. Click **Devices > Identity Servers > Edit > SAML 2.0 > Service Provider** or **Identity Provider**> **Options** > **New**.
5. Select **SAML2 POST DEFLATE TRUSTEDPROVIDERS** in **Property Type** and **enter** trusted provider's name, metadata URI, or provider ID in **Property Value**. You can specify multiple trusted providers in a comma separated format. These are the trusted providers who expect SAML2 POST messages in deflated format. In other words, this provider has to send deflated SAML2 POST messages to the listed trusted providers.
6. Click **OK**.
7. Restart Identity Server by using this command: `/etc/init.d/novell-idp restart`.

Allowable IDP proxy indirections: Specifies whether the trusted identity provider can proxy the authentication request to another identity provider. A value of **None** specifies that the trusted identity provider cannot redirect an authentication request. Values 1-5 determine the number of times the request can be proxied. Select **Let IDP Decide** to let the trusted identity provider decide how many times the request can be proxied

Force authentication at Identity Provider: Specifies that the trusted identity provider must prompt users for authentication, even if they are already logged in.

Use automatic introduction: Attempts single sign-on to this trusted identity provider by automatically sending a passive authentication request to the identity provider. (A passive requests does not prompt for credentials.) The identity provider sends one of the following authentication responses:

- **When the federated user is authenticated at the identity provider:** The identity provider returns an authentication response indicating that the user is authenticated. The user gains access to the service provider without entering credentials (single sign-on).
- **When the federated user is not authenticated at the identity provider:** The identity provider returns an authentication response indicating that the user is not logged in. The user can then select a card for authentication, including the card for the identity provider. If the user selects the identity provider card, an authentication request is sent to the identity provider. If the credentials are valid, the user is also authenticated to the service provider.

IMPORTANT: Enable the **Use automatic introduction** option only when you are confident the identity provider will be up. If the server is down and does not respond to the authentication request, the user gets a page-cannot-be-displayed error. Local authentication is disabled because the browser is never redirected to the login page.

This option must be enabled only when you know the identity provider is available 99.999% of the time or when the service provider is dependent upon this identity provider for authentication.

---

5 Click **OK** twice, then update Identity Server.

Understanding Comparison Contexts

When a service provider makes a request for an identity provider to authenticate a user, the authentication request can contain a class or type and a comparison context. The identity provider uses these to determine which authentication procedure to execute. There are four comparison contexts:

- **Exact:** Indicates that the class or type specified in the authentication statement must be an exact match to at least one contract.

  For example, when the comparison context is set to exact, the identity provider uses the URI in the request to find an authentication procedure. If an exact URI match is found, the user is prompted for the appropriate credentials. If an exact match is not found, the user is denied access.

- **Better:** Indicates the contract that must be stronger than the class or type specified in the authentication statement.

  If the identity provider is a NetIQ Identity Server, Identity Server first finds the specified class or type and its assigned authentication level. It then uses this information to find a contract that matches the conditions. For example if the authentication level is set to 1 for the class or type, the identity provider looks for a contract with an authentication level that is higher than 1. If one is found, the user is prompted for the appropriate credentials. If more than one is found, the user is presented with the matching cards and is allowed to select the contract. If a match is not found, the user is denied access.

- **Minimum:** Indicates that the contract must be as strong as the class or type specified in the authentication statement.

  If the identity provider is a NetIQ Identity Server, Identity Server first finds the specified class or type and its assigned authentication level. It then uses this information to find a contract that matches the conditions. For example if the authentication level is set to 1 for the class or type, the identity provider looks for a contract with an authentication level of 1 or higher. If one is found, the user is prompted for the appropriate credentials. If more than one is found, the user is presented with the matching cards and is allowed to select the contract. If a match is not found, the user is denied access.

- **Maximum:** Indicates that contract must as strong as possible without exceeding the strength of at least one of the authentication contexts specified.

  If the identity provider is a NetIQ Identity Server, Identity Server first finds the specified classes or types and their assigned authentication levels. It then uses this information to find a contract that matches the conditions. For example if the authentication level is set to 1 for some types and 3 for other types, the identity provider looks for contracts with an authentication level of 3. If a match or matches are found, the user is presented with the appropriate login prompts. If there are no contracts defined with a authentication level of 3, the identity provider looks for a match with an authentication level of 2, or if necessary, level 1. It cannot search below the lowest level of class in the authentication request or higher than the highest level of a class in the authentication request.

When you configure the authentication request, you specify the comparison context for a type or a contract.

Configuring the SAML 2.0 Authentication Response

After you create a trusted service provider, you can configure how your Identity Server responds to authentication requests from the service provider.

1 In the Administration Console, click **Devices > Identity Servers > Edit > SAML 2.0 > [Service Provider] > Authentication Response**.

2 Select the binding method.

  If the request from the service provider does not specify a response binding, you need to specify a binding method to use in the response. Select **Artifact** to provide an increased level of security by using a back-channel means of communication between the two servers. Select **Post** to use HTTP redirection for the communication channel between the two servers. If you select **Post**, you might want to require the signing of the authentication requests. See “Configuring the General Identity Provider Settings” on page 146.

  NOTE: The post binding can be configured to be sent as a compressed option. Perform the following steps to achieve this:

  1. In the Administration Console, click **Devices > Identity Servers > Edit** > **Options** > **New**.
  2. Select **IS SAML2 POST INFLATE** in **Property Type** and **true** in **Property Value**. This provider will receive deflated SAML2 POST messages from its trusted providers.
  3. Click **OK**.
  4. Click **Devices > Identity Servers > Edit > SAML 2.0 > Service Provider** or **Identity Provider**> **Options** > **New**.

5. Select **SAML2 POST DEFLATE TRUSTEDPROVIDERS** in **Property Type** and **enter** trusted provider's name, metadata URI, or provider ID in **Property Value**. You can specify multiple trusted providers in a comma separated format. These are the trusted providers who expect SAML2 POST messages in deflated format. In other words, this provider has to send deflated SAML2 POST messages to the listed trusted providers.
6. Click **OK**.
7. Restart Identity Server by using this command: `/etc/init.d/novell-idp restart`.

---

3 Specify the identity formats that Identity Server can send in its response. Select the box to choose one or more of the following:

- **Persistent:** Specifies that a persistent identifier, which is written to the directory and remains intact between sessions, can be sent.
- **Transient:** Specifies that a transient identifier, which expires between sessions, can be sent.
- **E-mail:** Specifies that an e-mail attribute can be used as the identifier.
- **Kerberos:** Specifies that a Kerberos token can be used as the identifier.
- **X509:** Specifies that an X.509 certificate can be used as the identifier.
- **Unspecified:** Specifies that an unspecified format can be used and any value can be used. The service provider and the identity provider need to agree on the value that is placed in this identifier.

4 Use the **Default** button to select the name identifier that Identity Server must send if the service provider does not specify a format.

If you select E-mail, Kerberos, x509, or unspecified as the default format, you must also select a value. See Step 5.

IMPORTANT: If you have configured the identity provider to allow a user matching expression to fail and still allow authentication by selecting the **Do nothing** option, you need to select **Transient identifier format** as the default value. Otherwise the users who fail the matching expression are denied access. To view the identity provider configuration, see “Defining User Identification for Liberty and SAML 2.0” on page 416.

---

5 Specify the value for the name identifier.

The persistent and transient formats are generated automatically. For the others, you can select an attribute. The available attributes depend upon the attributes that you have selected to send with authentication (see “Configuring the Attributes Obtained at Authentication” on page 156). If you do not select a value for the E-mail, Kerberos, X509, or Unspecified format, a unique value is automatically generated.

6 To specify that this Identity Server must authenticate the user, disable the **Use proxied requests** option. When the option is disabled and Identity Server cannot authenticate the user, the user is denied access.

When this option is enabled, Identity Server checks to see if other identity providers can satisfy the request. If one or more can, the user is allowed to select which identity provider performs the authentication. If a proxied identity provider performs the authentication, it sends the response to Identity Server. Identity Server then sends the response to the service provider.

7 Click **OK** twice, then update Identity Server.

Defining Options for SAML 2.0

OIOSAML enables service providers to use external authentication services, implements single sign-on across disparate systems, and establishes a foundation for federated identity management. OIOSAML enables reuse of authentication services and consistent application of security technology.

You can implement the Single Logout Profile of OIOSAML. This profile enables you to logout from all service providers whose session originate from a particular identity provider. For using this profile, you must use a front channel binding.

This section includes the following topics:

- "Defining Options for a SAML 2.0 Identity Provider" on page 438
- "Defining Options for a SAML 2.0 Service Provider" on page 441

Defining Options for a SAML 2.0 Identity Provider

1 In Administration Console, click **Devices > Identity Servers > Servers > Edit > SAML 2.0 > Identity Provider** > **Options**.

2 Select the required options:

OIOSAML Compliance: Select this option to make the identity provider OIOSAML compliant.

Enable Front Channel Logout: Select this option to enable a enable a service provider to initiates a logout at the identity provider by using the HTTP Redirect method.

3 Click **New** to set SAML properties for an identity provider. The following table lists the available properties:

NOTE: Access Manager 4.2 onwards, configuring the following options through files is deprecated. You must configure these option by using the Administration Console.

Property Type	Property Value
Extensions	Specify the value in this format: <samlp:Extensions>. This value is sent in the authentication request to this identity provider.
SAML ASSERTION INCLUDE MILLISECS	Select true to get SAML requests for this identity provider including the timestamp in millisecond in IssueInstant.
SAML2 ATTRIBUTE CONSUMING INDEX	Select the value of AttributeConsumingServiceIndex in SAML requests to this identity provider from the specified integer value.
SAML2 AVOID CONSENT	Select true to not include `Consent` as part of the SAML 2.0 request to this identity provider.
SAML2 AVOID ISPASSIVE	If you select true, IsPassive is not included in SAML 2.0 request to this identity provider.
SAML2 AVOID NAMEIDPOLICY	If you select true, NameIDPolicy is not included in a SAML 2.0 request to this identity provider.
SAML2 AVOID PROTOCOLBINDING	If you select true, ProtocolBinding is not included in a SAML 2.0 request to this identity provider.
SAML2 AVOID PROXYCOUNT	If you select true, ProxyCount is not included in a SAML 2.0 request to this identity provider.
SAML2 AVOID SIGN AND VALIDATE ASSERTIONS TRUSTED PROVIDERS	If you select true, the cluster will accept SAML 2.0 POST responses from this provider when the response is signed and assertion is not.
SAML2 CHANGE ISSUER	Specify the provider ID to be sent as issuer in the SAML requests to this identity provider. The value is in format {SPProviderID}->{issuer name}. {SPProviderID} will be replaced by the actual provider ID of the service provider. This will set the issuer of SAML 2.0 requests to the issuer name specified here. For example, https://nam.rtreyresearch.net:8443/nidp/saml2/metadata->https://saml.mariagerfjord.dk:8443/nidp/saml2/metadata.
SAML2 POST DEFLATE TRUSTEDPROVIDERS	If you select true, the cluster will send deflated post messages to this provider.
SAML2 SEND ACS INDEX	Select true to send AssertionConsumerServiceIndex with AuthnRequest to this identity provider.
SAML2 SEND ACS URL	Select true to send AssertionConsumerServiceURL with AuthnRequest to this identity provider.
SAML2 SIGN METHODDIGEST SHA256	Set the value to true to use SHA256 algorithm as signing algorithm for assertions.
OTHER	Specify Property Name and Value if you want to configure any other property for this identity provider.

4 Click **OK** > **Apply**.

Sample XML File When All SAML Options Are Set to True

<samlp:AuthnRequest xmlns:samlp="urn:oasis:names:tc:SAML:2.0:protocol" xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion" AssertionConsumerServiceIndex="2" ForceAuthn="false" ID="id5R6u1JFtay7eK.il97Q3eRl34u8" IssueInstant="2013-01-18T06:11:26Z" Version="2.0">

<saml:Issuer> https://nam.rtreyresearch.net:8443/nidp/saml2/metadata</saml:Issuer>

</samlp:AuthnRequest>

Sample XML File When All SAML Options Are Set to False

<samlp:AuthnRequest xmlns:samlp="urn:oasis:names:tc:SAML:2.0:protocol"xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion" AssertionConsumerServiceIndex="0" AttributeConsumingServiceIndex="2"Consent="urn:oasis:names:tc:SAML:2.0:consent:unavailable" ForceAuthn="false" ID="idoeZTKq7FOs5MsCigBBCwp30lqD0" IsPassive="false"IssueInstant="2013-01-23T05:25:32Z"ProtocolBindingProtocolBinding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST"Version="2.0">

<saml:Issuer> https://saml.mariagerfjord.dk:8443/nidp/saml2/metadata</saml:Issuer>

<samlp:NameIDPolicyAllowCreate="true"Format="urn:oasis:names:tc:SAML:2.0:nameid-format:persistent"SPNameQualifier="https://nam.rtreyresearch.net:8443/nidp/saml2/metadata"/><samlp:Scoping ProxyCount="5"/>

</samlp:AuthnRequest>

Defining Options for a SAML 2.0 Service Provider

You can use Access Manager as an identity provider for several service providers.You can configure a specific authentication contract that is required for a service provider. If you have configured more than one authentication contract for a service provider, the contract with minimum level is selected.

When providing authentication to a service provider, Identity Server ensures that the user is authenticated by the required contract. When a user is not authenticated or when a user is authenticated, but the authenticated contracts do not satisfy the required contracts, user is prompted to authenticate with the required contract. This is called step up authentication.

If no required contract is configured, then the default contract is executed.

NOTE: For SAML 2.0, this step up authentication is supported for Intersite Transfer Service (for both identity provider initiated and service provider initiated requests). For Liberty, it works only for identity provider initiated requests.

Perform the following steps to define options for a SAML 2.0 service provider:

1. In Administration Console, click **Devices > Identity Servers > Servers > Edit > SAML 2.0 > Service Provider** > **Options**.
2. Select **OIOSAML Compliance** to make the service provider OIOSAML compliant. The OIOSAML attribute set is automatically populated with the required attributes to send with authentication after selecting this check box.
3. Select the required step up authentication contracts from the **Available contracts** list and move them to the **Selected contracts** list. This is to provide the step up authentication for the service provider.

NOTE: Only the contract that is configured first in the **Selected contracts** list will be executed. This is applicable only for SAML 2.0.

4 Click **New** to set SAML properties for a service provider. The following table lists the available properties:

NOTE: Access Manager 4.2 onwards, configuring the following options through files is deprecated. You must configure these option by using the Administration Console.

Property Type	Description
SAML ASSERTION INCLUDE MILLISECS	Select true to get SAML responses for this service provider including the timestamp in millisecond in IssueInstant.
SAML2 AVOID AUDIENCE RESTRICTION	Select true to avoid sending the audience restriction information with assertion to this service provider.
SAML2 AVOID CONSENT	Select true to not include `Consent` as part of the SAML 2.0 request.
SAML2 AVOID SIGN AND VALIDATE ASSERTIONS TRUSTED PROVIDERS	If you select true, the cluster will sign SAML 2.0 POST responses (excluding the assertion) for this provider.
SAML2 AVOID SPNAMEQUALIFIER	Select true to not include `SPNAMEQUALIFIER` in `NAMEIDENTIFER` in the assertion.
SAML2 AVOID SPNAMEQUALIFIER TO	Select true to send SPNAMEQUALIFIER in NAMEIDENTIFER with the assertion.
SAML2 NAMEID ATTRIBUTE NAME	Specify the LDAP attribute name that will be sent in the name identifier in a SAML response for this service provider.
SAML2 POST DEFLATE TRUSTEDPROVIDERS	If you select true, the cluster will send deflated post messages to this provider.
SAML2 POST SIGN RESPONSE TRUSTEDPROVIDERS	If you select true, the identity provider will sign the entire SAML 2.0 response for this service provider.
SAML2 REQUEST IGNORE AUTHCONTEXT	If you select true, the identity provider ignores any specific authentication available in a SAML request from this service provider.
SAML2 SIGN METHODDIGEST SHA256	If you select true, assertion will use the SHA 256 algorithm as a hashing algorithm for this service provider.
OTHER	Specify Property Name and Value if you want to configure any other property for this service provider.

5 Click **OK** > **Apply**.

Defining Session Synchronization for the A-Select SAML 2.0 Identity Provider

If a user session is active on the service provider, the service provider periodically sends session synchronization to Identity Server to maintain the session. You must configure the properties for the session synchronization between the service provider and the target identity provider.

1. In the Administration Console, click **Devices > Identity Servers > Servers > Edit > Liberty or SAML 2.0 > Identity Provider** > **Options.**
2. Click **New**.
3. Select **Other** in **Property Type**.
4. Specify the following values:

   Property Name: `config.aselect.sessionsync.enabled`

   Property Value: `true`
5. For session synchronization, add two options, one to enable the session synchronization and the other to provide the URL to which synchronization message must be sent.

   The session synchronization message is sent from the Access Manager service provider to the A-Select identity provider, in tandem with the Access Gateway ESP's activity update. The session synchronization message is sent only if the user session is active at the Access Gateway portal, which is the ESP to the Access Manager service provider. If you log in directly to the Access Manager service provider, even if the session is active, the session synchronization message is not sent to the A-Select identity provider.
6. Click **OK,** then update Identity Server.

Configuring the Liberty or SAML 2.0 Session Timeout

When you are active on a session on the service provider and a timeout occurs, the service provider initiates a logout.You can configure this timeout by using the `web.xml` parameter in the Access Gateway ESP, then ESP initiates a logout message to the Access Manager service provider over the SOAP back channel when the timeout is reached. After the service provider receives this message, it creates a SAML 2.0 logout request to the remote identity provider over SOAP.

To send session timeout message:

1. In the Administration Console, click **Devices** > **Access Gateways** > **Edit** > **Reverse Proxy / Authentication** > **ESP Global Options**.
2. Remove the pound (#) symbol before notifysessionTimetoIDP and set the value as true.

   ESP sends a ESP session timeout message. After timeout, the service provider sends a `samlp:LogoutRequest xmlns:samlp="urn:oasis:names:tc:SAML:2.0:protocol` request to the remote identity provider.
3. Restart Tomcat on each Identity Server in the cluster by using the following command:

   /etc/init.d/novell-idp restart

Session Termination

If you set the session synchronization between the Service Provider and remote Identity Provider, then the remote Identity Provider never sends the logout request to the active Service Provider.

Modifying the Authentication Card for Liberty or SAML 2.0

When you create an identity provider, you must also configure an authentication card. After it is created, you can modify it.

1 In the Administration Console, click **Devices > Identity Servers > Edit > [Protocol] > [Identity Provider] > Authentication Card.**

2 Modify the values in one or more of the following fields:

ID: If you have need to reference this card outside of the user interface, specify an alphanumeric value here. If you do not assign a value, Identity Server creates one for its internal use. The internal value is not persistent. Whenever Identity Server is rebooted, it can change. A specified value is persistent.

Text: Specify the text that is displayed on the card to the user. This value, in combination with the image, must identify to the users, which provider they are logging into.

Image: Specify the image to be displayed on the card. Select the image from the drop-down list. To add an image to the list, click **<Select local image>**.

Show Card: Determine whether the card is shown to the user, which allows the user to select and use the card for authentication. If this option is not selected, the card is only used when a service provider makes a request for the card.

NOTE: Do not disable the **Show Card** option for default contracts.

Passive Authentication Only: Select this option if you do not want Identity Server to prompt the user for credentials. If Identity Server can fulfill the authentication request without any user interaction, the authentication succeeds. Otherwise, it fails.

Satisfies Contract: Select the required contracts from the **Available contracts** list and move them to the **Satisfies contract** list.

If Access Manager Identity provider is unable to execute the requested authentication contract, it looks for the configured external Identity provider. This happens when **Satisfiable by External Contract** option is enabled which satisfies the incoming authentication (contract) request. If the match is found, Identity provider lists all the satisfiable contracts to select the appropriate contract. If only a single match is found, Identity provider redirects it to an external contract.

If the local Identity Provider is able to authenticate using a local contract, which is satisfiable by an external provider, then the preference is first given to the local contract along with the other authentication cards also listed.

To configure the contract matching criteria, see Allowable Class in “Configuring Authentication Contracts” on page 289.

3 Click **OK** twice, then update Identity Server.

Configuring Multiple SAML 2.0 Service Providers on the Same Host for a Single SAML Identity Provider

When the same Access Manager server hosts more than one SAML service provider and federate with another Access Manager acting as an identity provider for these service providers, Access Manager must send different sets of attributes in SAML 2.0 assertions to these service providers.

Perform the following steps to create multiple service providers on the same Access Manager host:

1. To create multiple service providers from the same identity provider metadata, manually modify the identity provider's metadata's entityID for each service provider. You can import the metadata text that was edited into the Access Manager configuration to create service providers with different entity IDs.

   For more information about how to create a SAML 2.0 service provider, see "Creating a Trusted Service Provider" on page 152.
2. In the Administration Console of the SAML 2.0 identity provider, click **Devices** > **Identity Servers** > **Servers** > **Edit** > **SAML 2.0** > **Service Provider** > **Options** > **New**.
3. Set the SAML2 AVOID AUDIENCE RESTRICTION property to true. Setting this property to true avoids audience restriction in the SAML 2.0 assertion.
4. To avoid the spnamequalifier attribute in nameidentifier of the assertion, do the following:
   - **4a** In Administration Console of the SAML 2.0 service provider, click **Devices** > **Identity Servers** > **Servers** > **Edit** > **SAML 2.0** > **Service Provider** > **Options** > **New.**
   - **4b** Set the SAML2 AVOID SPNAMEQUALIFIER TO property to true.
   - **4c** Click **OK**.
5. Restart Identity Server.

NOTE: This is possible only when the identity provider and service providers are deployed on Access Manager.

Configuring Active Directory Federation Services with SAML 2.0 for Single Sign-On

This section describes step-by-step instructions for configuring a basic identity federation deployment between Microsoft Active Directory Federation Services 2.0 (AD FS 2.0) and Access Manager by using the Security Assertion Markup Language (SAML) 2.0 protocol, specifically its Web Browser SSO Profile and HTTP POST binding.

You can configure AD FS 2.0 as the claims provider and Access Manager as the relying party, or you can configure Access Manager as the claims provider and AD FS 2.0 as the relying party or service provider.

- "Prerequisites and Requirements" on page 446
- "Configuring Access Manager as a Claims or Identity Provider and AD FS 2.0 as Relying Party or Service Provider" on page 447
- "Configuring AD FS 2.0 as the Claims or Identity Provider and Access Manager as the Relying Party or Service Provider" on page 454
- "AD FS 2.0 Basics" on page 457
- "Debugging AD FS 2.0" on page 458

Prerequisites and Requirements

- Two servers, one to host AD FS 2.0 and the other to host Access Manager.
- AD FS 2.0 is deployed.
- ADFS 2.0 with WIF is deployed.

  The test deployment that was created in the AD FS 2.0 Federation with a Windows Identity Foundation (WIF) (http://go.microsoft.com/fwlink/?LinkId=193997) application is used as starting point for this deployment. A single Windows Server 2012 instance (fsweb.contoso.com) is used to host both the AD FS 2.0 federation server and a WIF sample application. It presumes the availability of a Contoso.com domain, in which fsweb.contoso.com is a member server. The same computer can act as the domain controller and federation server in the test deployments.
- ADFS 2.0 with SharePoint 2010 is deployed.

  The test deployment that was created in the Configuring SharePoint 2010 AAM applications with AD FS 2.0 (http://technet.microsoft.com/en-us/library/gg295319.aspx) is used as starting point for this deployment. A single Windows Server 2012 instance (fsweb.contoso.com) is used to host the AD FS 2.0 federation server and a Windows Server 2012 instance (SP2010) is used to host the SharePoint 2010 application. It presumes the availability of a Contoso.com domain, in which fsweb.contoso.com is a member server. The same computer can act as the domain controller and federation server in the test deployments.
- Access Manager is deployed.

  The Access Manager environment in this deployment is hosted by a fictitious company called nam.example.com. Only Identity Server component of Access Manager is required for this federation. For more information about installation and deployment of Access Manager, see NetIQ Access Manager Appliance 4.2 Installation and Upgrade Guide.

NOTE: You can download the evaluation version of Access Manager from NetIQ's download portal.

Environment

- Access Manager 4.x.x.
- SUSE Linux Enterprise Server (SLES) 11 SP4 64-bit or a higher version.

IP Connectivity

Ensure that the Access Manager (nam.example.com) and AD FS 2.0 (fsweb.contoso.com) systems have IP connectivity between them. The Contoso.com domain controller, if it is running on a separate computer, does not require IP connectivity to the Access Manager system. If the Access Manager firewall is set up, open the ports required for Identity Server to communicate with the Administration Console.

For more information about these ports, see "Setting Up Firewalls" in the *NetIQ Access Manager Appliance 4.2 Installation and Upgrade Guide*.

For HTTPS communication, the Access Manager Identity Server uses TCP 8443 by default. Your browsers need to access this port when using the HTTP POST Binding. Or, you can change this port to 443 by using iptables.

For back-channel communication with cluster members, you need to open two consecutive ports for the cluster, such as 7801 and 7802. The initial port (7801) is configurable. See "Configuring a Cluster with Multiple Identity Servers" on page 55.

All federation servers (AD FS and Access Manager) need access to a reliable Network Time Protocol (NTP) time source.

Name Resolution

The hosts file on the AD FS 2.0 computer (fsweb.contoso.com) is used to configure name resolution of the partner federation servers and sample applications.

Clock Synchronization

Federation events have a short time to live (TTL). To avoid errors based on time-outs, ensure that both computers have their clocks synchronized.

NOTE: On SLES 11 SP1 64-bit or a higher version, use the command `sntp -P no -p pool.ntp.org` to synchronize time with the Internet time server.

Configuring Access Manager as a Claims or Identity Provider and AD FS 2.0 as Relying Party or Service Provider

This section explains how to configure a setup in which an Access Manager user gets federated access to the WIF sample application or SharePoint 2010 through AD FS 2.0. This setup uses the SAML 2.0 POST profile.

- “Configuring Access Manager” on page 447
- “Configuring AD FS 2.0” on page 449
- “Example Scenario: Access Manager as the Claims Provider and AD FS 2.0 as the Relying Party” on page 453

Configuring Access Manager

- Using Metadata to Add a New Service Provider Connection
- Exporting the Identity Provider Metadata to a File

NOTE: To deploy this identity federation, create a new contract with the “urn:oasis:names:tc:SAML:2.0:ac:classes:Password” URI and with the name password form method. Configure this contract as the default contract.

Using Metadata to Add a New Service Provider Connection

The first step in configuring Access Manager is to use the AD FS metadata to add a service provider for Access Manager.

- Getting the AD FS 2.0 Metadata
- Using the Metadata to Add a New Service Provider Connection
- Adding an AD FS Server Trusted Certificate
- Creating an Attribute Set in Access Manager
- Configuring the Service Provider in Access Manager

Getting the AD FS 2.0 Metadata

1. Access the AD FS server metadata URL at https://<<ADFS (hostname or IP)/FederationMetadata/2007-06/FederationMetadata.xml.
2. Save the AD FS metadata file.
3. Open the saved AD FS metadata file in Notepad, WordPad, or in any XML editor.

4 Remove the <RoleDescriptor> tags from the metadata. For example, remove the following tags:

```
<RoleDescriptor xsi:type="fed:ApplicationServiceType"
protocolSupportEnumeration=http://..................... ……> ……..</
RoleDescriptor>

  <RoleDescriptor xsi:type="fed:SecurityTokenServiceType"
protocolSupportEnumeration=http://.....  ……..> </RoleDescriptor>
```

5 Save the changes.

Using the Metadata to Add a New Service Provider Connection

1 In the Access Manager Administration Console, click **Devices** > **Identity Server** > **Edit** > **SAML 2.0**.
2 Click **New** > **Add Service Provider**.
3 In the **Name** field, specify a name by which you want to refer to the provider.
4 Select **Metadata Text** from the **Source** list.
5 Paste the copied AD FS metadata that you saved in Step 5 on page 448 into the **Text** field.
6 Click **Next** > **Finish**.
7 Update Identity Server.

Adding an AD FS Server Trusted Certificate

1 Download the certificate authority (CA) certificate from the AD FS server.
2 In the Access Manager Administration Console, click **Security** > **Certificates** > **Trusted Roots**.
3 Click **Import**.
4 Specify a name for the certificate and browse for the ADFS certificate.
5 Click **OK**.
6 Click **Uploaded AD FS CA**.
7 Click **Add to Trusted Store** and select **config store**.
8 Update Identity Server.

Creating an Attribute Set in Access Manager

1 In the Access Manager Administration Console, click **Devices** > **Identity Servers** > **Shared Settings** > **Attribute Sets** > click **New**.
2 Provide the attribute set name as adfs-attributes.
3 Click **Next** with the default selections.
4 In the **Create Attribute Set** section, click **New**.
5 Select **ldapattribute mail** from the **Local Attribute** list.
6 Specify **emailaddress** in the **Remote attribute** field.
7 Select **http://schemas.xmlsoap.org/ws/2005/05/identity/claims/** from the **Remote namespace** list.
8 Click **OK**.
9 Click **New**.
10 Select **All Roles** from the **Local Attribute** list.
11 Specify roles in the **Remote Attribute** field.

12 Select **http://schemas.xmlsoap.org/ws/2005/05/identity/claims/** from the **Remote namespace** list.

13 Click **OK**.

14 Update Identity Server.

Configuring the Service Provider in Access Manager

1 In the Access Manager Administration Console, select the ADFS service provider in the **SAML 2.0** tab.

2 Click **Authentication Response**.

3 Select **Binding to POST**.

4 Specify the name identifier format default value and select **unspecified** along with the defaults.

5 Click **Attributes**.

6 Select **adfs-attributes** from the **Attribute Set** list.

7 Select the required attributes to be sent with authentication. For example, the mail and cn attributes.

8 Click **OK**.

9 Update Identity Server.

Exporting the Identity Provider Metadata to a File

Access https://<<Identity server IP / dns name>>:8443/nidp/saml2/metadata in a browser and save the page as an XML file, such as`nam_metadata.xml`. AD FS 2.0 uses this file to automate the setup of the Access Manager Claims Provider instance.

Configuring AD FS 2.0

- Using Metadata to Add Claims Provider
- Editing Claim Rules for the Claims Provider Trust
- Editing Claim Rules for the WIF Sample Application
- Editing Claim Rules for the SharePoint 2010 Application
- Disabling CRL Checking in the Linux Identity Server

Using Metadata to Add Claims Provider

You need to use the metadata import capabilities of AD FS 2.0 to create the Example.com claims provider. The metadata includes the public key that is used to validate security tokens signed by Access Manager.

Using Metadata to Add a Relying Party

1 In AD FS 2.0, in the console tree, right-click the **Claims Provider Trusts** folder, then click **Add Claims Provider Trust** to start the Add Claims Provider Trust Wizard.

2 Click **Start**.

3 On the Select Data Source page, select **Import data about the claims provider from a file**.

4 In the **Federation metadata file location** field, click **Browse**.

5 Navigate to the location where you saved `nam_metadata.xml`, click **Open**, then click **Next**.

6 On the Specify Display Name page, type `NAM Example`.

7 Click **Next** > **Next** > **Close**.

Editing Claim Rules for the Claims Provider Trust

The following claim rule describes how the data from Access Manager is used in the security token that is sent to the WIF sample application or SharePoint 2010.

1. In AD FS 2.0, click **Relying Party Trusts**, right click **WIF Sample App**, and then click **Edit Claim Rules**.

   or

   In the AD FS 2.0 center pane, under **Claims Provider Trusts**, right-click **NAM Example**, then click **Edit Claim Rules**.
2. On the **Acceptance Transform Rules** tab, click **Add Rule**.
3. On the Select Rule Template page, select the **Pass Through or Filter an Incoming Claim** option.
4. Click **Next**.
5. On the Configure Claim Rule page, use the following values:

Name	Value
Claim rule name	Name ID Rule
Incoming claim type	Name ID
Incoming name ID format	Unspecified

6. Select the **Pass through all claim values** option and click **Finish**.
7. Click **Add Rule**.
8. On the Select Rule Template page, select the **Pass Through or Filter an Incoming Claim** option.
9. Click **Next**.
10. On the Configure Claim Rule page, under **Claim rule name**, use the following values:

Name	Value
Claim rule name	Name Rule
Incoming claim type	Name

11. Leave the **Pass through all claim values** option selected and click **Finish**.
12. To acknowledge the security warning, click **Yes**.
13. Click **OK**.
14. Click **Add Rule**.
15. On the Select Rule Template page, select the **Pass Through or Filter an Incoming Claim** option.
16. Click **Next**.
17. On the Configure Claim Rule page, in the **Claim rule name** field, use the following values:

Name	Value
Claim rule name	Email Rule
Incoming claim type	E-Mail Address

18. Leave the **Pass through all claim values** option selected and click **Finish**.

19 To acknowledge the security warning, click **Yes**.

20 Click **OK**.

Editing Claim Rules for the WIF Sample Application

At this point, incoming claims have been received at AD FS 2.0, but rules that describe what to send to the WIF sample application have not yet been created. You need to edit the existing claim rules for the sample application to take into account the new Access Manager external claims provider.

1 In AD FS 2.0, click **Relying Party Trusts**.

2 Right-click **WIF Sample App**, then click **Edit Claim Rules**.

3 On the **Issuance Transform Rules** tab, click **Add Rule**.

4 On the Select Rule Template page, click **Pass Through or Filter an Incoming Claim** > **Next**.

5 On the Configure Claim Rule page, enter the following values:

Name	Value
Claim rule name	Pass Name Rule
Incoming claim type	Name

6 Leave the **Pass through all claim values** option selected, then click **Finish**.

7 On the **Issuance Transform Rules** tab, click **Add Rule**.

8 On the Select Rule Template page, click **Pass Through or Filter an Incoming Claim**.

9 Click **Next**.

10 On the Configure Claim Rule page, enter the following values:

Name	Value
Claim rule name	Pass Name ID Rule
Incoming claim type	Name ID
Incoming Name ID format	Unspecified

11 Leave the **Pass through all claim values** option selected, then click **Finish**.

12 Click **OK**.

NOTE: If you changed the rules while federating AD FS 2.0 with the WIF sample application, ensure that you add the Permit All Users issuance rules back to the WIF sample application. See Step 6: – Change Rules in the AD FS 2.0 Federation with a WIF Application Step-by-Step Guide.

Or, as an alternative, add a new Permit or Deny Users Based on an Incoming Claim rule allowing incoming Name ID = john@example.com to access the application.

Editing Claim Rules for the SharePoint 2010 Application

At this point, incoming claims have been received at AD FS 2.0, but the rules that describe what to be sent to the SharePoint 2010 application have not yet been created. You need to edit the existing claim rules for the SharePoint 2010 application, which is added as relying party to ADFS 2.0, to configure the new Access Manager external claims provider.

Editing the Claim Rules for the SharePoint 2010 Application

1. In AD FS 2.0, click **Relying Party Trusts**.
2. Right-click **SP2010**, then click **Edit Claim Rules.**
3. On the **Issuance Transform Rules** tab, click **Add Rule**.
4. On the Select Rule Template page, click **Pass Through or Filter an Incoming Claim** > **Next**.
5. On the Configure Claim Rule page, enter the following values:

Name	Value
Claim rule name	Pass eMail Rule
Incoming claim type	Email Address

6. Leave the **Pass through all claim values** option selected and click **Finish**.

Using Certificates and Certificate Revocation Lists

For security reasons, production federation deployments require the use of digitally signed security tokens, and optionally allows encryption of the security token contents. Self-signed private key certificates, which are generated from inside the AD FS 2.0 and Access Manager products, are used for signing security tokens.As an alternative, organizations can use a private key certificate that is issued by a certificate authority (CA) for signing and encryption. The primary benefit of using CA-issued certificates is the ability to check for possible certificate revocation against the certificate revocation list (CRL) from the issuing CA. Also, to avoid the untrusted certificate messages in browsers, the trusted root certificate of the CA must also be imported into your browsers. Many well-known CA's trusted roots are included with common browsers. Using one of these existing CAs to mint your certificates also prevents the untrusted certificate messages.

In AD FS 2.0 and in Access Manager, CRL checking is enabled by default for all partner connections, if the certificate being used by the partner includes a CRL Distribution Point (CDP) extension. This has implications in federation deployments between Access Manager and AD FS 2.0:

- If a signing/encryption certificate provided by one side of a federation includes a CDP extension, that location must be accessible by the other side's federation server. Otherwise, CRL checking fails, resulting in a failed access attempt. The CDP extensions are added by default to certificates that are issued by Active Directory Certificate Services (AD CS) in Windows Server 2012.
- If the signing/encryption certificate does not include a CDP extension, no CRL checking is performed by AD FS 2.0 or Access Manager.

Disabling CRL Checking in the Linux Identity Server

1. Modify `/opt/novell/nam/idp/conf/tomcat.conf` and add `JAVA_OPTS="${JAVA_OPTS} -Dcom.novell.nidp.serverOCSPCRL=false"`
2. To apply the changes, restart Identity Server by running the `/etc/init.d/novell-idp restart` command.

Disabling CRL Checking in AD FS 2.0

1. Click **Start** > **Administrative Tools** > **Windows PowerShell Modules**.
2. Enter the following command at the PowerShell command prompt:

```
set-ADFSClaimsProviderTrust -TargetName "NAM Example"
-SigningCertificateRevocationCheck None
```

NOTE: You can make many configuration changes to AD FS 2.0 by using the Windows PowerShell command line and scripting environment. For more information, see the AD FS 2.0 Windows PowerShell Administration section of the AD FS 2.0 Operations Guide and the AD FS 2.0 Cmdlets Reference (http://go.microsoft.com/fwlink/?LinkId=177389).

Example Scenario: Access Manager as the Claims Provider and AD FS 2.0 as the Relying Party

- "Accessing the WIF Sample Application" on page 453
- "Accessing the SharePoint 2010 Application" on page 453

Accessing the WIF Sample Application

In this scenario, John from Example.com accesses the Contoso WIF sample application.

NOTE: Clear all the cookies in the Internet Explorer on the AD FS 2.0 computer (fsweb.contoso.com). To clear the cookies, click **Tools** > **Internet Options** > **Delete** under **Browsing History**, and then select cookies for deletion.

1. On the AD FS 2.0 computer, open a browser window, then navigate to https://fsweb.contoso.com/ClaimsAwareWebAppWithManagedSTS/default.aspx.

   The first page prompts you to select your organization from a list.
2. Select **NAM Example**, then click **Continue** to sign in.

   When only one Identity Provider is available, AD FS 2.0 forwards the request to that Identity Provider by default.
3. The NAM login page appears. Type the user name john, type the password test, then click **Login**.

Accessing the SharePoint 2010 Application

The user's email ID is used as the mapped attribute to access the SharePoint 2010 application. Assume that a user is created in the NetIQ Identity Server. The email ID configured for this user is namuser1@namidp.com.

NOTE: Clear all the cookies in the Internet Explorer on the AD FS 2.0 computer (fsweb.contoso.com). To clear the cookies, click **Tools** > **Internet Options** > **Delete** under **Browsing History**, then select cookies for deletion.

1. Ensure that an email ID has been configured for the user in the Access Manager user store.

   For this example, use namuser1@namidp.com.
2. Access the SharePoint 2010 application.

   The user is redirected to AD FS 2.0.
3. Select **NetIQ Identity Server**.

   The user is redirected to the NAM IDP nidp page for authentication.
4. Provide namuser1 as the username and password.

   After authentication, the user is redirected to the SharePoint application.

Configuring AD FS 2.0 as the Claims or Identity Provider and Access Manager as the Relying Party or Service Provider

This section explains how to configure an application through AD FS 2.0 that gets federated access to an application by using Access Manager. The setup uses the SAML 2.0 POST profile.

- "Configuring Access Manager" on page 454
- "Configuring AD FS 2.0" on page 455

Configuring Access Manager

The AD FS metadata is used to add an Identity Provider to Access Manager.

- "Getting the AD FS 2.0 Metadata" on page 454
- "Using the Metadata to Add a New Identity Provider Connection" on page 454
- "Adding the AD FS Server Trusted Certificate" on page 455
- "Configuring the Identity Provider in Access Manager" on page 455

Getting the AD FS 2.0 Metadata

1. Access the AD FS server metadata by going to `https://<<ADFS hostname or IP/FederationMetadata/2007-06/FederationMetadata.xml`
2. Save the AD FS metadata data.
3. Open the AD FS metadata file in Notepad, WordPad, or an XML editor).
4. Remove the `<RoleDescriptor>` tags from the metadata.

   For example, remove the following tags:

   ```
   "<RoleDescriptor xsi:type="fed:ApplicationServiceType"
              protocolSupportEnumeration=http://..... ……> …….</
   RoleDescriptor>

   "<RoleDescriptor xsi:type="fed:SecurityTokenServiceType"
               protocolSupportEnumeration=http://.....  ……> </RoleDescriptor>
   ```

5. Save the changes.

Using the Metadata to Add a New Identity Provider Connection

1. In the Access Manager Administration Console, select **Devices** > **Identity Server**.
2. Click **Edit**.
3. Select **SAML 2.0**.
4. Click **New** > **Identity Provider**.
5. Specify the name as **ADFS** in the **Name** field.
6. Select **Metadata Text** from the **Source** list.
7. Paste the copied ADFS metadata that you saved in Step 5 on page 454 into the **Text** field.
8. Click **Next**.
9. Specify an alphanumeric value that identifies the card in the **ID** field.
10. Specify the image to be displayed on the card in the **Image** field.
11. Update Identity Server.

Adding the AD FS Server Trusted Certificate

1. Retrieve the AD FS server's CA trusted root certificate.
2. In the Access Manager Administration Console, select **Security** > **Certificates**.
3. Select **Trusted Roots**.
4. Click **Import**.
5. Specify the certificate name, and browse for the AD FS certificate authority.
6. Click **OK**.
7. Click **uploaded AD FS CA**.
8. Click **Add to Trusted Store and select config store**.
9. Update Identity Server.

Configuring the Identity Provider in Access Manager

1. Select the **AD FS Identity Provider** in the **SAML 2.0** tab.
2. Click **Authentication Card** > **Authentication Request**.
3. Select **Response Protocol Binding to POST**.
4. Select **NAME Identifier Format as Transient**.
5. Click **OK**.
6. Update Identity Server.

Configuring AD FS 2.0

- "Using the Metadata to Add a Relying Party" on page 455
- "Editing Claim Rules for a Relying Party Trust" on page 456
- "Disabling the Certificate Revocation List" on page 456
- "AD FS 2.0 Encryption Strength" on page 457

Using the Metadata to Add a Relying Party

The metadata import capability of AD FS 2.0 is used to create a relying party. The metadata includes the public key that is used to validate security tokens signed by Access Manager.

1. In AD FS 2.0, right-click the **Relying Party Trusts** folder, then click **Add Relying Party Trust** to start the Add Relying Party Trust Wizard.
2. Click **Start**.
3. On the Select Data Source page, select **Import data about the claims provider from a file**.
4. In the **Federation metadata file location** section, click **Browse**.
5. Navigate to the location where you saved `nam_metadata.xml` earlier, select the file, then click **Open** > **Next**.
6. On the Specify Display Name page, specify NAM Example.
7. Click **Next** > **Next** > **Close**.

Editing Claim Rules for a Relying Party Trust

The data from AD FS is used in the security token that is sent to Access Manager.

1. The Edit Claim Rules dialog box must already be open. If not, in the AD FS 2.0 center pane, under **Relying Party Trusts**, right-click **NAM Example**, then click**Edit Claim Rules**.

2 On the **Issuance Transform Rules** tab, click **Add Rule**.

3 On the Select Rule Template page, leave the **Send LDAP Attributes as Claims** option selected, then click **Next**.

4 On the Configure Claim Rule page, specify `Get attributes` in the **Claim rule name** field.

5 Select **Active Directory** from the **Attribute Store** list.

6 In the **Mapping of LDAP attributes** section, create the following mappings:

LDAP Attribute	Outgoing Claim Type
User-Principal-Name	UPN
E-Mail-Address	E-Mail Address

7 Click **OK**.

8 Click **Apply** > OK.

9 On the **Issurance Transform Rules** tab, click **Add Rules**.

10 On the Select Rule Template page, select **Transform an Incoming Claim**, then click **Next**.

11 On the Configure Claim Rule page, use the following values:

Name	Value
Claim rule name	Mapping To Transient Name Identifier
Incoming Claim Type	UPN
Outgoing Claim Type	Name ID
Outgoing name ID format	Transient Identifier

12 Select **Pass Through All Claims**, then click **OK**.

13 Click **Apply** > **OK**.

Disabling the Certificate Revocation List

For more information about signing and encryption certificates, see “Using Certificates and Certificate Revocation Lists” on page 452.

Disabling the CRL Checking Option in the Linux Identity Provider

Disabling the CRL Checking Option in AD FS 2.0

1 Click **Start** > **Administrative Tools** > **Windows PowerShell Modules**.

2 Enter the following command at the PowerShell command prompt:

```
set-ADFSRelyingPartyTrust -TargetName "NAM Example"
-SigningCertificateRevocationCheck None
```

AD FS 2.0 Encryption Strength

In AD FS 2.0, encryption of the outbound assertions is enabled by default. Assertion encryption occurs for any relying party or service provider for which AD FS 2.0 possesses an encryption certificate. AD FS 2.0 uses 256-bit Advanced Encryption Standard (AES) keys or AES-256 for encryption. In contrast, Failing to reconcile these conflicting defaults can result in the failed SSO attempts. To resolve this issue, disable the encryption in AD FS 2.0.

1. In AD FS 2.0, click **Start** > **Administrative Tools** > **Windows PowerShell Modules**.
2. Enter the following command in at the PowerShell command prompt:

   ```
   set-ADFSRelyingPartyTrust -TargetName "NAM Example"
   -EncryptClaims $False
   ```

AD FS 2.0 Basics

- "Configuring the Token-Decrypting Certificate" on page 457
- "Adding CA Certificates to AD FS 2.0" on page 457

Configuring the Token-Decrypting Certificate

1. Open the AD FS 2.0 Management tool, then click **Start** > **Administrative Tools** > **AD FS 2.0 Management**.
2. In the left pane, expand the **Service** folder and click **Certificates**.
3. In the **Certificates** section, select **Add Token-Decrypting Certificate**.
4. (Conditional) If you see an error prompting you to run certain commands during the token-decrypting process, run the following PowerShell commands:

   ```
   Add-PSSnapin Microsoft.Adfs.PowerShell
   ```

   ```
   Set-ADFSProperties -AutoCertificateRollover $false
   ```

   These commands allow you to select other certificates. The certificate must be installed on the server. The certificates are configured on the IIS Manager.
5. Click **Start** > **Administrative Tools** > **Internet Information Services (IIS) Manager**.
6. Click **ServerName**.
7. Click **Server Certificates** in the IIS section.

Adding CA Certificates to AD FS 2.0

1. In Windows, **Start** > **Run** > **mmc**.
2. Attach snapshot certificates as service.
3. Select **AD FS**.
4. Import the CA certificate to trusted authorities.

Debugging AD FS 2.0

1 In the **Event Viewer**, click **Applications** > **AD FS**. You can access the troubleshooting help at Troubleshooting certificate problems with AD FS 2.0 (http://technet.microsoft.com/en-us/library/adfs2-troubleshooting-certificate-problems%28WS.10%29.aspx).

Power Shell Commands Help:

- Using Windows PowerShell for AD FS2.0 (http://technet.microsoft.com/en-us/library/adfs2-help-using-windows-powershell%28WS.10%29.aspx)
- AD FS 2.0 for Windows PowerShell Examples (http://technet.microsoft.com/en-us/library/adfs2-powershell-examples%28WS.10%29.aspx)

Configuring SAML 1.1

This section explains how to use the SAML 1.1 protocol to set up the trust with internal and external identity providers, service providers, and Embedded Service Providers (ESPs). Topics include:

- "Configuring a SAML 1.1 Profile" on page 458
- "Creating a SAML 1.1 Service Provider" on page 459
- "Creating a SAML 1.1 Identity Provider" on page 459
- "Configuring Communication Security for SAML 1.1" on page 459
- "Editing a SAML 1.1 Identity Provider's Metadata" on page 459
- "Editing a SAML 1.1 Service Provider's Metadata" on page 459
- "Configuring the SAML 1.1 Authentication Response" on page 459
- "Defining Options for SAML 1.1 Service Provider" on page 460
- "Modifying the Authentication Card for SAML 1.1" on page 460

Configuring a SAML 1.1 Profile

You can configure the methods of communication that are available at the server for requests and responses sent between providers. These settings affect the metadata for the server and must be determined prior to publishing to other sites.

Profiles control what methods of communication are available at the server for the SAML 1.1 protocol. These settings affect the metadata for the server and must be determined prior to publishing to other sites. If you have set up trusted providers, and then modify these profiles, the trusted providers need to reimport the metadata from this Identity Server.

1 In the Administration Console, click **Devices > Identity Servers > Edit > SAML 1.1 > Profiles**.

2 Configure the following fields:

Login: Specifies the communication channel when the user logs in. Select one or more of these methods for the identity provider and the identity consumer:

- The Artifact binding provides an increased level of security by using the back channel for communication between the two servers during authentication.

- The Post method uses HTTP redirection to accomplish communication between servers.

  The Post method is enabled by default and you are not able to modify the default settings.The Post profile creates a metadata that includes only a Post binding on the Service Provider. If you have the default setup, then always both Artifact and Post options are enabled. If both the options are enabled, then by default Artifact binding is used. If Artifcact binding is disabled or removed, only Post method is used.

Source ID: Displays the hexadecimal ID generated by the Identity Server for the SAML 1.1 service provider. This is a required value when establishing trust with a service provider.

3 Click **OK**, then update the Identity Server.

4 (Conditional) If you have set up trusted providers and have modified the profile, these providers need to reimport the metadata from this Identity Server.

Creating a SAML 1.1 Service Provider

See “Creating a Trusted Service Provider” on page 152.

Creating a SAML 1.1 Identity Provider

See “Creating a Trusted Identity Provider” on page 150.

Configuring Communication Security for SAML 1.1

Liberty and SAML 1.1 have the same security options for the SOAP back channel for both identity and service providers. See “Configuring Communication Security for Liberty” on page 463

Editing a SAML 1.1 Identity Provider’s Metadata

See “Editing a SAML 1.1 Identity Provider’s Metadata” on page 160.

Editing a SAML 1.1 Service Provider’s Metadata

See “Editing a SAML 1.1 Service Provider’s Metadata” on page 161.

Configuring the SAML 1.1 Authentication Response

You can specify the name identifier and its format when the Identity Server sends an authentication response. You can also restrict the use of the assertion.

When an identity provider sends an assertion, the assertion can be restricted to an intended audience. The intended audience is defined to be any abstract URI in SAML 1.1. The URL reference can also identify a document that describes the terms and conditions of audience membership.

1 In the Administration Console, click **Devices > Identity Servers > Edit > SAML 1.1 > [Service Provider]** > **Authentication Response**.

2 To specify a name identifier format, select one of the following:

- **E-mail:** Specifies that an e-mail attribute can be used as the identifier.
- **X509:** Specifies that an X.509 certificate can be used as the identifier.

- **Unspecified:** Specifies that an unspecified format can be used and any value can be used. The service provider and the identity provider need to agree on what value is placed in this identifier.

3 To specify the format of the name identifier, select an attribute.

The available attributes depend upon the attributes that you have selected to send with authentication (see the Attributes page for the service provider).

4 To configure an audience, click **New**.

5 Specify the **SAML Audience URL** value.

The Provider ID, which can be used for the audience, is displayed on the Edit page for the metadata.

6 Click **OK** twice, then update the Identity Server.

Defining Options for SAML 1.1 Service Provider

For more information about Options, see "Defining Options for a SAML 2.0 Service Provider" on page 441

1 In Administration Console, click **Devices > Identity Servers > Servers > Edit > SAML 1.1 > Service Provider** > **Options**.

2 Select the required step up authentication contracts from the **Available Contracts** list and move them to the **Selected Contracts** list. These selected contracts will be used to provide the step up authentication for the service provider.

3 Click **OK**.

Modifying the Authentication Card for SAML 1.1

When you create an identity provider, you must also configure an authentication card. After it is created, you can modify it.

1 In the Administration Console, click **Devices > Identity Servers > Edit > SAML 1.1 > [Identity Provider] > Authentication Card.**

2 Modify the values in one or more of the following fields:

ID: If you have need to reference this card outside of the user interface, specify an alphanumeric value here. If you do not assign a value, the Identity Server creates one for its internal use. The internal value is not persistent. Whenever the Identity Server is rebooted, it can change. A specified value is persistent.

Text: Specify the text that is displayed on the card to the user. This value, in combination with the image, must identify to the users, which provider they are logging into.

Login URL: Specify an Intersite Transfer Service URL.The URL has the following format, where idp.sitea.novell.com is the DNS name of the identity provider, idp.siteb.novell.com is the name of the service provider, and idp.siteb.novell.com:8443/nidp/app specifies the URL that you want to users to access after a successful login.

NOTE: The PID in the login URL must exactly match the entity ID specified in the metadata.

```
https://idp.sitea.novell.com:8443/nidp/saml/idpsend?PID=https://
idp.siteb.novell.com:8443/nidp/saml/metadata&TARGET=https://
idp.siteb.novell.com:8443/nidp/app
```

For more information, see “Specifying the Intersite Transfer Service URL for the Login URL Option” on page 166.

If your identity provider is a NetIQ Identity Server and you know the ID specified for the target, you can use the following simplified format for the Login URL:

```
<URL for site a>?id=<ID of target>
```

```
https://idp.sitea.novell.com:8443/nidp/saml/idpsend?id=206test
```

The target and the target ID are specified in the service provider configuration at the identity provider. See “Configuring an Intersite Transfer Service Target for a Service Provider” on page 169.

Image: Specify the image to be displayed on the card. Select the image from the drop-down list. To add an image to the list, click **<Select local image>**.

Show Card: Determine whether the card is shown to the user, which allows the user to select and use the card for authentication. If this option is not selected, the card is only used when a service provider makes a request for the card.

3 Click **OK** twice, then update the Identity Server.

Configuring Liberty

This section explains how to use the Liberty protocol to set up the trust with internal and external identity providers, service providers, and Embedded Service Providers (ESPs). Topics include:

- “About Liberty” on page 461
- “Configuring a Liberty Profile” on page 462
- “Creating a Liberty Service Provider” on page 463
- “Creating a Liberty Identity Provider” on page 463
- “Configuring Communication Security for Liberty” on page 463
- “Configuring a Liberty Authentication Request” on page 464
- “Configuring the Liberty Authentication Response” on page 466
- “Defining Options for Liberty Service Provider” on page 466
- “Defining Options for Liberty Identity Provider” on page 467
- “Configuring the Session Timeout” on page 467
- “Modifying the Authentication Card” on page 467

About Liberty

The Liberty Alliance is a consortium of business leaders with a vision to enable a networked world in which individuals and businesses can more easily conduct transactions while protecting the privacy and security of vital identity information.

To accomplish its vision, the Liberty Alliance established an open standard for federated network identity through open technical specifications. In essence, this open standard is a structured version of the Security Assertions Markup Language, commonly referred to as SAML, with the goal of accelerating the deployment of standards-based single sign-on technology.

For general information about the Liberty Alliance, visit the Liberty Alliance Project Web site (http://www.projectliberty.org/).

Liberty resources, including specifications, white papers, FAQs, and presentations, can be found at the Liberty Alliance Resources Web site (http://www.projectliberty.org/liberty/resource_center/).

The following table provides links to specific Liberty Alliance specifications:

Table 5-15 *Liberty Alliance Links*

Liberty Specification	Location
Liberty Alliance Project Overview	Liberty Alliance Project Overview (http://www.projectliberty.org/)
Liberty White Papers	Papers (http://www.projectliberty.org/liberty/resource_center/papers)
Identity Federation Specifications	Liberty ID-FF 1.2 Specification (http://www.projectliberty.org/resources/specifications.php#box1)
Web Service Framework Specifications	Liberty ID-WSF 1.1 Specifications (http://www.projectliberty.org/resources/specifications.php#box2a)
Liberty Profile Service Specifications	Liberty Alliance ID-SIS 1.0 Specifications (http://www.projectliberty.org/resources/specifications.php#box3)
OASIS Standards (SAML)	Oasis Standards (http://www.oasis-open.org/specs/index.php#samlv2.0)

Configuring a Liberty Profile

You can configure the methods of communication that are available at the server for requests and responses sent between providers. These settings affect the metadata for the server and must be determined prior to publishing to other sites.

The profile specifies what methods of communication are available at the server for the Liberty protocol. These settings affect the metadata for the server and must be determined prior to publishing to other sites. If you have set up trusted providers, and then modify these profiles, the trusted providers need to reimport the metadata from this Identity Server.

1. In the Administration Console, click **Devices > Identity Servers > Edit > Liberty > Profiles**.
2. Configure the following fields for identity providers and service providers:

   Login: Specifies whether to support Artifact or Post binding for login. Select one or more of the following for the identity provider and the service provider:

   - The **Artifact** binding provides an increased level of security by using a back channel means of communication between the two servers during authentication.
   - The **Post** method uses HTTP redirection to accomplish communication between the servers.

   Single Logout: Specifies the communication method to use when the user logs out. Typically, you select both of these options, which enables the identity provider or service provider to accept both HTTP and SOAP requests. SOAP is used if both options are selected, or if the service provider has not specified a preference.

   - **HTTP:** Uses HTTP 302 redirects or HTTP GET requests to communicate logout requests from this identity site to the service provider.

- **SOAP:** Uses SOAP over HTTP messaging to communicate logout requests from this identity provider to the service provider.

Federation Termination: Specifies the communication channel to use when the user selects to defederate an account. Typically, you select both of these options, which enables the identity provider or service provider to accept both HTTP and SOAP requests. SOAP is the default setting if the service provider has not specified a preference.

- **HTTP:** Uses HTTP 302 redirects to communicate federation termination requests from this server.
- **SOAP:** Uses SOAP back channel over HTTP messaging to communicate logout requests from this server

Register Name: Specifies the communication channel to use when the provider supplies a different name to register for the user. Typically, you select both of these options, which enables the identity provider or service provider to accept both HTTP and SOAP requests. SOAP is the default setting if the service provider has not specified a preference.

- **HTTP:** Uses HTTP 302 redirects to communicate federation termination requests from this server.
- **SOAP:** Uses SOAP back channel over HTTP messaging to communicate logout requests from this server.

3 Click **OK**, then update the Identity Server.

4 (Conditional) If you have set up trusted providers and have modified the profile, these providers need to reimport the metadata from this Identity Server.

Creating a Liberty Service Provider

See "Creating a Trusted Service Provider" on page 152.

Creating a Liberty Identity Provider

See "Creating a Trusted Identity Provider" on page 150.

Configuring Communication Security for Liberty

Liberty and SAML 1.1 have the same security options for the SOAP back channel for both identity and service providers. You cannot configure the trust relationship of the SOAP back channel for the Identity Server and its Embedded Service Providers.

1 In the Administration Console, click **Devices > Identity Servers > Edit** > **[Protocol]**.

For the protocol, select either Liberty or SAML 1.1.

2 Click the name of a provider.

3 On the Trust page, fill in the following field:

Name: Specify the display name for this trusted provider. The default name is the name you entered when creating the trusted provider.

For an Embedded Service Provider, the **Name** option is the only available option on the Trust page.

The **Security** section specifies how to validate messages received from trusted providers over the SOAP back channel. Both the identity provider and the service provider in the trusted relationship must be configured to use the same security method.

4 Select one of the following security methods:

Message Signing: Relies upon message signing using a digital signature.

Mutual SSL: Specifies that this trusted provider provides a digital certificate (mutual SSL) when it sends a SOAP message.

SSL communication requires only the client to trust the server. For mutual SSL, the server must also trust the client. For the client to trust the server, the server's certificate authority (CA) certificate must be imported into the client trust store. For the server to trust the client, the client's CA certificate must be imported into the server trust store.

Basic Authentication: Specifies standard header-based authentication. This method assumes that a name and password for authentication are sent and received over the SOAP back channel.

- **Send:** The name and password to be sent for authentication to the trusted partner. The partner expects this password for all SOAP back-channel requests, which means that the name and password must be agreed upon.
- **Verify:** The name and password used to verify data that the trusted provider sends.

5 Click **OK** twice.

6 Update the Identity Server.

Configuring a Liberty Authentication Request

You can configure how the Identity Server creates an authentication request for a trusted identity provider. When users authenticate, they can be given the option to federate their account identities with the preferred identity provider. This process creates an account association between the identity provider and service provider that enables single sign-on and single log-out.

The authentication request specifies how you want the identity provider to handle the authentication process so that it meets the security needs of the Identity Server.

1 In the Administration Console, click **Devices > Identity Servers > Edit > Liberty > [Identity Provider] > Authentication Card** > **Authentication Request**.

2 Configure the federation options:

Allow Federation: Determines whether federation is allowed. The federation options that control when and how federation occurs can only be configured if the identity provider has been configured to allow federation.

- **After authentication:** Specifies that the federation request can be sent after the user has authenticated (logged in) to the service provider. When you set only this option, users must log in locally, then they can federate by using the **Federate** option on the card in the Login page of the Access Manager User Portal. Because the user is required to authenticate locally, you do not need to set up user identification.
- **During authentication:** Specifies whether federation can occur when the user selects the authentication card of the identity provider. Typically, a user is not authenticated at the service provider when this selection is made. When the identity provider sends a response to the service provider, the user needs to be identified on the service provider to complete the federation. If you enable this option, ensure that you configure a user identification method. See "Selecting a User Identification Method for Liberty or SAML 2.0" on page 417.

3 Select one of the following options for the **Requested By** option:

Do not specify: Specifies that the identity provider can send any type of authentication to satisfy a service provider's request, and instructs a service provider to not send a request for a specific authentication type or contract.

Use Types: Specifies that authentication types must be used.

Select the types from the **Available types** field to specify which type to use for authentication between trusted service providers and identity providers. Standard types include Name/Password, Secure Name/Password, X509, Token, and so on.

Use Contracts: Specifies that authentication contracts must be used.

Select the contract from the **Available contracts** list. For a contract to appear in the **Available contracts** list, the contract must have the **Satisfiable by External Provider** option enabled. To use the contract for federated authentication, the contract's URI must be the same on the identity provider and the service provider. For information about contract options, see "Configuring Authentication Contracts" on page 289.

Most third-party identity providers do not use contracts.

4 Configure the options:

Response protocol binding: Select **Artifact** or **Post** or **None**. Artifact and Post are the two methods for transmitting assertions between the authenticating system and the target system.

If you select **None**, you are letting the identity provider determine the binding.

Identity Provider proxy redirects: Specifies whether the trusted identity provider can proxy the authentication request to another identity provider. A value of **None** specifies that the trusted identity provider cannot redirect an authentication request. Values 1-5 determine the number of times the request can be proxied. Select **Configured on IDP** to let the trusted identity provider decide how many times the request can be proxied.

Force authentication at Identity Provider: Specifies that the trusted identity provider must prompt users for authentication, even if they are already logged in.

Use automatic introduction: Attempts single sign-on to this trusted identity provider by automatically sending a passive authentication request to the identity provider. (A passive requests does not prompt for credentials.) The identity provider sends one of the following authentication responses:

- **When the federated user is authenticated at the identity provider:** The identity provider returns an authentication response indicating that the user is authenticated. The user gains access to the service provider without entering credentials (single sign-on).
- **When the federated user is not authenticated at the identity provider:** The identity provider returns an authentication response indicating that the user is not logged in. The user can then select a card for authentication, including the card for the identity provider. If the user selects the identity provider card, an authentication request is sent to the identity provider. If the credentials are valid, the user is also authenticated to the service provider.

IMPORTANT: Enable the **Use automatic introduction** option only when you are confident the identity provider will be up. If the server is down and does not respond to the authentication request, the user gets a page-cannot-be-displayed error. Local authentication is disabled because the browser is never redirected to the login page.

This option must be enabled only when you know the identity provider is available 99.999% of the time or when the service provider is dependent upon this identity provider for authentication.

---

5 Click **OK** twice, then update the Identity Server.

Configuring the Liberty Authentication Response

After you create a trusted service provider, you can configure how your Identity Server responds to authentication requests from the service provider.

1. In the Administration Console, click **Devices > Identity Servers > Edit > Liberty > [Service Provider] > Authentication Response**.
2. Select the binding method.

   If the request from the service provider does not specify a response binding, you need to specify a binding method to use in the response. Select **Artifact** to provide an increased level of security by using a back-channel means of communication between the two servers. Select **Post** to use HTTP redirection for the communication channel between the two servers. If you select **Post**, you might want to require the signing of the authentication requests. See "Configuring the General Identity Provider Settings" on page 146.
3. Specify the identity formats that the Identity Server can send in its response. Select the **Use** box to choose one or more of the following:
   - **Persistent Identifier Format:** Specifies a persistent identifier that federates the user profile on the identity provider with the user profile on the service provider. It remains intact between sessions.
   - **Transient Identifier Format:** Specifies that a transient identifier, which expires between sessions, can be sent.

   If the request from the service provider requests a format that is not enabled, the user cannot authenticate.
4. Use the **Default** button to specify whether a persistent or transient identifier is sent when the request from the service provider does not specify a format.
5. To specify that this Identity Server must authenticate the user, disable the **Use proxied requests** option. When the option is disabled and the Identity Server cannot authenticate the user, the user is denied access.

   When this option is enabled, the Identity Server checks to see if other identity providers can satisfy the request. If one or more can, the user is allowed to select which identity provider performs the authentication. If a proxied identity provider performs the authentication, it sends the response to the Identity Server. The Identity Server then sends the response to the service provider.
6. Enable the **Provide Discovery Services** option if you want to allow the service provider to query the Identity Server for a list of its Web Services. For example, when the option is enabled, the service provider can determine whether the Web Services Framework is enabled and which Web Service Provider profiles are enabled.
7. Click **OK** twice, then update the Identity Server.

Defining Options for Liberty Service Provider

NetIQ Access Manager can be used as an identity provider for several service providers.You can configure a specific authentication contract that is required for a Service provider. If more than one authentication contract is configured for a service provider, the contract having minimum level will be selected.

When providing authentication to a service provider, the identity server ensures that the user is authenticated by the required contract. When a user is not authenticated or when user is authenticated, but the authenticated contracts do not satisfy the required contracts, user will be prompted to authenticate with required contract. This is called step up authentication.

If no required contract is configured, then the default contract is executed.

NOTE: This step up authentication is supported only for Intersite Transfer Service (identity provider initiated) requests on Liberty and works for both identity and service provider initiated requests for SAML 2.0.

To Define Options for Liberty Service Provider

1. In the Administration Console, click **Devices > Identity Servers > Servers > Edit > Liberty > Service Provider** > **Options**.
2. Select the required step up authentication contracts from the **Available contracts** list and move them to the **Selected contracts** list. This is to provide the step up authentication for the service provider.
3. Click **OK**.

Defining Options for Liberty Identity Provider

1. In Administration Console, click **Devices > Identity Servers > Servers > Edit > Liberty or SAML 2.0 > Identity Provider** > **Options**.
2. **Enable Front Channel Logout:** After this option is enabled, a service provider initiates a logout at the identity provider by using the HTTP Redirect method.

Configuring the Session Timeout

See “Configuring the Liberty or SAML 2.0 Session Timeout” on page 443.

Modifying the Authentication Card

See “Modifying the Authentication Card for Liberty or SAML 2.0” on page 444.

Configuring Liberty Web Services

A Web service uses Internet protocols to provide a service. It is an XML-based protocol transported over SOAP, or a service whose instances and data objects are addressable via URIs.

Access Manager consists of several elements that comprise Web services:

- **Web Service Framework:** Manages all Web services. The framework defines SOAP header blocks and processing rules that enable identity services to be invoked via SOAP requests and responses.
- **Web Service Provider:** An entity that provides data via a Web service. In Access Manager, Web service providers host Web service profiles, such as the Employee Profile, Credential Profile, Personal Profile, and so on.
- **Web Service Consumer:** An entity that uses a Web service to access data. Web service consumers discover resources at the Web service provider, and then retrieve or update information about a user, or on behalf of a user. Resource discovery among trusted partners is necessary because a user might have many kinds of identities (employee, spouse, parent, member of a group), as well as several identity providers (employers or other commercial Web sites).

- **Discovery Service:** The service assigned to an identity provider that enables a Web Service Consumer to determine which Web service provider provides the required resource.
- **LDAP Attribute Mapping:** Access Manager's solution for mapping Liberty attributes with established LDAP attributes.

This section describes the following topics:

- "Web Services Framework" on page 468
- "Managing Web Services and Profiles" on page 469
- "Configuring Credential Profile Security and Display Settings" on page 475
- "Customizing Attribute Names" on page 476
- "Configuring the Web Service Consumer" on page 477
- "Mapping LDAP and Liberty Attributes" on page 477

For additional resources about the Liberty Alliance specifications, visit the Liberty Alliance Specification (http://www.projectliberty.org/resources/specifications.php) page.

Web Services Framework

The Web Services Framework page lets you edit and manage all the details that pertain to all Web services. This includes the framework for building interoperable identity services, permission-based attribute sharing, identity service description and discovery, and the associated security mechanisms.

1. In the Administration Console, click **Devices > Identity Servers** > *Edit* > **Liberty > Web Service Framework**.
2. Fill in the following fields:

   Enable Framework: Enables Web Services Framework.

   Axis SOAP Engine Settings: Axis is the SOAP engine that handles all Web service requests and responses. Web services are deployed using XML-based files known as Web service deployment descriptors (WSDD). On startup, Access Manager automatically creates the server-side and client-side configuration for Axis to handle all enabled Web services.

   If you need to override this default configuration, use the **Axis Server Configuration WSDD XML** field and the **Axis Client Configuration WSDD XML** field to enter valid WSDD XML. If either or both of these controls contain valid XML, then Access Manager does not automatically create the configuration (server or client) on startup.
3. Click **OK**.

Managing Web Services and Profiles

After a service has been discovered and data has been received from a trusted identity provider, the Web service consumer can invoke the service at the Web service provider. A Web service provider is the hosting or relying entity on the server side that can make access control decisions based on this data and upon its business practices and preferences.

1. In the Administration Console click **Identity Servers** > *Edit* > **Liberty > Web Service Provider**.
2. Select one of the following actions

   New: To create a new Web service, click **New**. This activates the Create Web Service Wizard. You can create a new profile only if you have deleted one.

   Delete: To delete an existing profile, select the profile, then click **Delete**.

   Enable: To enable a profile, select the profile, then click **Enable**.

Disable: To disable a profile, select the profile, then click **Disable**.

Edit a Policy: To edit the policy associated with a profile, click the **Policy** link. For configuration information, see "Editing Web Service Policies" on page 473.

Edit a profile: To edit a profile, click the name of a profile. For information about configuring the details, see "Modifying Service and Profile Details for Employee, Custom, and Personal Profiles" on page 470 and "Modifying Details for Authentication, Discovery, LDAP, and User Interaction Profiles" on page 471.

For information about modifying the description, see "Editing Web Service Descriptions" on page 472.

The Identity Server comes with the following Web service profile types:

Authentication Profile: Allows the system to access the roles and authentication contracts in use by current authentications. This profile is enabled by default so that Embedded Service Providers can evaluate roles in policies. This profile can be disabled. When it is disabled, all devices assigned to use this Identity Server cluster configuration cannot determine which roles a user has been assigned, and the devices evaluate policies as if the user has no roles.

WARNING: Do not delete this profile. In normal circumstances, this profile is used only by the system.

Credential Profile: Allows users to define information to keep secret. It uses encryption to store the data in the directory the user profile resides in.

Custom Profile: Used to create custom attributes for general use.

Discovery: Allows requesters to discover where the resources they need are located. Entities can place resource offerings in a discovery resource, allowing other entities to discover them. Resources might be a personal profile, a calendar, travel preferences, and so on.

Employee Profile: Allows you to manage employment-related information and how the information is shared with others. A company address book that provides names, phones, office locations, and so on, is an example of an employee profile.

LDAP Profile: Allows you to use LDAP attributes for and general use.

Personal Profile: Allows you to manage personal information and to determine how to share that information with others. A shopping portal that manages the user's account number is an example of a personal profile.

User Interaction: Allows you to set up a trusted user interaction service, used for identity services that must interact with the resource owner to get information or permission to share data with another Web service consumer. This profile enables a Web service consumer and Web service provider to cooperate in redirecting the resource owner to the Web service provider and back to the Web service consumer.

3 Click **OK**.

4 On the Servers page, update the Identity Server.

Modifying Service and Profile Details for Employee, Custom, and Personal Profiles

The settings on the Details page are identical for the Employee, Custom, and Personal Profiles. This page allows you to specify the display name, resource ID encryption, and how the system reads and writes data.

1. In the Administration Console, click **Devices > Identity Servers** > *Edit* **> Liberty > Web Service Provider**.
2. Click **Custom Profile**, **Employee Profile**, or **Personal Profile**, depending on which profile you want to edit.
3. Click the **Details** tab (it is displayed by default).
4. Specify the general settings, as necessary:

   Display Name: The Web service name. This specifies how the profile is displayed in the Administration Console.

   Have Discovery Encrypt This Service's Resource Ids: Specifies whether the Discovery Service encrypts resource IDs. A resource ID is an identifier used by Web services to identify a user. The Discovery Service returns a list of resource IDs when a trusted service provider queries for the services owned by a given user. The Discovery Service has the option of encrypting the resource ID or sending it unencrypted.
5. Specify data location settings:

   Selected Read Locations: The list of selected locations from which the system reads attributes containing profile data. If you add multiple entries to this list, the system searches attributes in each location in the order you specify. When a match is found for an attribute, the other locations are not searched. Use the up/down and left/right arrows to control which locations are selected and the order in which to read them. Read locations can include:

   - **Configuration Datastore:** Liberty attribute values can be stored in the configuration store of the Administration Console. If your users have access to the User Portal, they can add values to a number of Liberty attributes.
   - **LDAP Data Mappings:** If you have mapped a Liberty attribute to an LDAP attribute in your user store, the values can be read from the LDAP user store. To create LDAP attribute maps, see "Mapping LDAP and Liberty Attributes" on page 477.
   - **Remote Attributes:** If you set up federation, the Identity Server can read attributes from these remote service providers. Sometimes, the service provider is set up to push a set of attribute values when the user logs in. These pushed attributes are cached, and the Identity Server can quickly read them. If a requested attribute has not been pushed, a request for the Liberty attribute is sent to remote service provider. This can be time consuming, especially if the user has federated with more than one remote service provider. **Remote Attributes** must always be the last item in this list.

   Available Read Locations: The list of available locations from which the system can read attributes containing profile data. Locations in this list are currently not being used.

   Selected Write Locations: The list of selected locations to write attribute data to. If you add multiple entries to this list, the system searches attributes in each location in the order you specify. When a match is found for an attribute, the other locations are not searched. Use the up/down and left/right arrows to control which locations are selected and the order in which they are selected.

   - **Configuration Datastore:** Liberty attribute values can be stored in the configuration store of the Administration Console. The Identity Server can write values to these attributes. If this location appears first in the list of **Selected Write Locations**, all Liberty attribute values are written to this location. If you want values written to the LDAP user store, the **LDAP Data Mappings** location must appear first in the list.

- **LDAP Data Mappings:** If you have mapped a Liberty attribute to an LDAP attribute in your user store, the Identity Server can write values to the attribute in the LDAP user store. To create LDAP attribute maps, see “Mapping LDAP and Liberty Attributes” on page 477.

Available Write Locations: The list of available locations to write attributes containing profile data. Locations in this list are currently not being used.

6 (Optional) Specify data model extensions.

Data Model Extension XML: The data model for some Web services is extensible. You can enter XML definitions of data model extensions in this field. Data model extensions hook into the existing Web service data model at predefined locations.

All schema model extensions reside inside of a schema model extension group. The group exists to bind model data items together under a single localized group name and description. Schema model extension groups can reside inside of a schema model extension root or inside of a schema model extension. There can only be one group per root or extension. Each root is hooked into the existing Web service data model. Multiple roots can be hooked into the same location in the existing Web service data model. This conceptual model applies to the structure of the XML that is required to define data model extensions.

See Appendix A, “Data Model Extension XML,” on page 1187 for more information.

7 Click **OK**, then click **OK** on the Web Service Provider page.

8 Update the Identity Server.

Modifying Details for Authentication, Discovery, LDAP, and User Interaction Profiles

This page allows you to specify information for Discovery, LDAP, and User Interaction Web service profiles. If you are creating a Web service type, this is Step 2 of the Create Web Service Wizard.

For conceptual information about profiles, see Managing Web Services and Profiles.

1 In the Administration Console, click **Devices**> **Identity Servers** > *Edit* > **Liberty** > **Web Service Provider** > **[Profile]**.

2 Click **Authentication, Discovery, LDAP,** or **User Interaction**, depending on which profile you want to edit.

3 Configure the following fields:

Display name: The Web service name. This specifies how the profile is displayed in the Administration Console.

Have Discovery Encrypt This Service’s Resource Ids: (Not applicable for the Discovery profile) Specifies whether the Discovery Service encrypts resource IDs. A resource ID is an identifier used by Web services to identify a user. The Discovery Service returns a list of resource IDs when a trusted service provider queries for the services owned by a given user. The Discovery Service has the option of encrypting the resource ID or sending it unencrypted. This ID is encrypted with the public key of the resource provider generated at installation. Encrypting resource IDs is turned off by default.

4 Click **OK**.

Editing Web Service Descriptions

All of the Description pages on each profile are identical. You can define how a service provider gains access to portions of the user’s identity information that can be distributed across multiple providers. The service provider uses the Discovery Service to ascertain the location of a specific identity service

for a user. The Discovery Service enables various entities to dynamically and securely discover a user's identity service, and it responds, on a permission basis, with a service description of the desired identity service.

1. In the Administration Console, click **Devices > Identity Servers** > *Edit* > **Liberty > Web Service Provider**.
2. Click the profile or service.
3. Click **Descriptions**.
4. Click the description name, or click **New**.
5. Fill in the following fields:

   Name: The Web Service Description name.

   Security Mechanism: (Required) Liberty uses channel security (TLS 1.0) and message security in conjunction with the security mechanism. Channel security addresses how communication between identity providers, service providers, and user agents is protected. For authentication, service providers are required to authenticate identity providers by using identity provider server-side certificates. Identity providers have the option to require authentication of service providers by using service provider client-side certificates.

   Message security addresses security mechanisms applied to the discrete Liberty protocol messages passed between identity providers, service providers, and user agents.

   Select the mechanism for message security. Message authentication mechanisms indicate which profile is used to ensure the authenticity of a message.

   - **X.509:** Used for message exchanges that generally rely upon message authentication as the principle factor in making decisions.
   - **SAML:** Used for message exchanges that generally rely upon message authentication as well as the conveyance and attestation of information.
   - **Bearer:** Based on the presence of the security header of a message. In this case, the bearer token is verified for authenticity rather than proving the authenticity of the message.
6. Under **Select Service Access Method**, select either **Brief Service Access Method** or **WSDL Service Access Method**.

   Brief Service Access Method: Provides the information necessary to invoke basic SOAP-over-HTTP-based service instances without using WSDL.

   - **EndPoint URL:** This is the SOAP endpoint location at the service provider to which Liberty SOAP messages are sent. An example of this for the Employee Profile is [BASEURL]/services/IDSISEmployeeProfile. If the service instance exposes an endpoint that is different from the logically generated concrete WSDL, you must use the WSDL URI instead.

     A WSF service description endpoint cannot contain double-byte characters.
   - **SOAP Action:** The SOAP action HTTP header required on HTTP-bound SOAP messages. This header can be used to indicate the intent of a SOAP message to the recipient.

   WSDL Service Access Method: Specify the method used to access the WSDL service. WSDL (Web Service Description Language) describes the interface of a Web service.

   - **Service Name Reference:** A reference name for the service.
   - **WSDL URI:** Provides a URI to an external concrete WSDL resource containing the service description. URIs need to be constant across all implementations of a service to enable interoperability.
7. Click **OK**.
8. Update the Identity Server configuration.

Editing Web Service Policies

Web Service policies are permission policies (query and modify) that govern how identity providers share end-user data with service providers. Administrators and policy owners (users) can control whether private information is always allowed to be given, never allowed, or must be requested.

As an administrator, you can configure this information for the policy owner, for specific service providers, or globally for all service providers. You can also specify what policies are displayed for the end user in the User Portal, and whether users are allowed to edit them.

1. In the Administration Console, click **Devices > Identity Servers** > *Edit* > **Liberty > Web Service Provider.**
2. Click the **Policy** link next to the service name.
3. Click the category you want to edit.

   All Trusted Providers: Policies that are defined by the service provider's ability to query and modify the particular Liberty attributes or groups of attributes for the Web service. When All Trusted Providers permissions are established, and a service provider needs data, the system first looks here to determine whether user data is allowed, never allowed, or must be asked for. If no solution is found in All Trusted Providers, the system examines the permissions established within the specific service provider.

   Owners: Policies that limit the end user's ability to modify or query data from his or her own profile. The settings you specify in the **Owner** group are reflected on the My Profile page in the User Portal. Portal users have the authority to modify the data items in their profiles. The data items include Liberty and LDAP attributes for personal identity, employment, and any customized attributes defined in the Identity Server configuration. Any settings you specify in the Administration Console override what is displayed in the User Portal. Overrides are displayed in the **Inherited** column.

   If you want the user to have Write permission for a given data item, and that data item is used in an LDAP Attribute Map, then you must configure the LDAP Attribute Map with Write permission.
4. On the All Service Policy page, select the policy's check box, then click **Edit Policy**.

   This lets you modify the parent service policy attribute. Any selections you specify on this page are inherited by child policies.

   Query Policy: Allows the service provider to query for the data on a particular attribute. This is similar to read access to a particular piece of data.

   Modify Policy: Allows the service provider to modify a particular attribute. This is similar to write access to a particular piece of data.

   Query and Modify: Allows you to set both options at once.
5. To edit child attributes of the parent, click the policy.

   In the following example, child attributes are inheriting Ask Me permission from the parent **Entire Personal Identity** attribute. The **Postal Address** attribute, however, is modified to never allow permission for sharing.

   If you click the **Postal Address** attribute, you can see that all of its child attributes have inherited the **Never Allow** setting. You can specify different permission attributes for **Address Type** (for example), but the inherited policy still overrides changes made at the child level, as shown below.

Postal Addresses

Postal Addresses

Edit Policy 6 Item(s)

Policy	Query Policy	Modify Policy	Inherited
Address Type	Always Allow	Always Allow	Never Allow : Never Allow
NickName	Ask Me	Ask Me	Never Allow : Never Allow
Localized NickNames	Ask Me	Ask Me	Never Allow : Never Allow
Comment	Ask Me	Ask Me	Never Allow : Never Allow
Postal Address	Ask Me	Ask Me	Never Allow : Never Allow
Postal Addresses Extensions	Ask Me	Ask Me	Never Allow : Never Allow

The interface allows these changes to simplify switching between configurations if, for example, you want to remove an inherited policy.

Inherited: Specifies the settings inherited from the parent attribute policy, when you view a child attribute. In the User Portal, settings displayed under **Inherited** are not modifiable by the user. At the top-level policy in the User Portal, the values are inherited from the settings in the Administration Console. Thereafter, inheritance can come from the service policy or the parent data item's policy.

Ask Me: Specifies that the service provider requests from the user what action to take.

Always Allow: Specifies that the identity provider always allows the attribute data to be sent to the service provider.

Never Allow: Specifies that the identity provider never allows the attribute data to be sent to the service provider.

When a request for data is received, the Identity Server examines policies to determine what action to take. For example, if a service provider requires a postal address for the user, the Identity Server performs the following actions:

- Checks the settings specified in **All Service Providers**.
- If no solution is found, checks for the policy settings configured for the service provider.

6 Click **OK** until the Web Service Provider page is displayed.

7 Click **OK**, then update the Identity Server as prompted.

Create Web Service Type

This page allows you to create a Web service profile type. This is Step 1 of the Create Web Service Wizard. Access Manager comes with several Web service profiles, but if you have deleted a profile type, you can create it again.

1 In the Administration Console, click **Devices > Identity Servers** > *Edit* > **Liberty > Web Service Provider > New.**

2 Select the Web service type from the drop-down list, then click **Next**.

3 Continue with one of the following:

- "Modifying Service and Profile Details for Employee, Custom, and Personal Profiles" on page 470.
- "Modifying Details for Authentication, Discovery, LDAP, and User Interaction Profiles" on page 471.

Configuring Credential Profile Security and Display Settings

On the Credential Profile Details page, you can specify whether this profile is displayed for end users, and determine how you control and store encrypted secrets. You can store and access secrets locally, on remote eDirectory servers that are running Novell SecretStore, or on a user store that has been configured with a custom attribute for secrets.

For more information about storing encrypted secrets, see the following:

- For information about how to configure Access Manager for secrets, see "Configuring a User Store for Secrets" on page 277.
- For general information about Novell SecretStore, see the *Novell SecretStore Administration Guide* (http://www.novell.com/documentation/secretstore33/pdfdoc/nssadm/nssadm.pdf).
- For information about creating shared secrets for Form Fill and Identity Injection policies, see "Creating and Managing Shared Secrets" on page 732.

To configure the Credential Profile:

1. In the Administration Console, click **Devices > Identity Servers** > *Edit* > **Liberty > Web Service Providers**.
2. Click **Credential Profile**.
3. On the Credential Profile Details page, fill in the following fields as necessary:

   Display name: The name you want to display for the Web service.

   Have Discovery Encrypt This Service's Resource Ids: Specify whether the Discovery Service encrypts the resource IDs. A resource ID is an identifier used by Web services to identify a user. The Discovery Service returns a list of resource IDs when a trusted service provider queries for the services owned by a given user. The Discovery Service has the option of encrypting the resource ID or sending it unencrypted. Encrypting resource IDs is disabled by default.
4. Under **Credential Profile Settings**, enable the following option if necessary:

   Allow End Users to See Credential Profile: Specify whether to display or hide the Credential Profile in the Access Manager User Portal. Profiles are viewed on the My Profile page, where the user can modify his or her profile.
5. Specify how you want to control and store secrets:

   5a. To locally control and store secrets, configure the following fields:

   Encryption Password Hash Key: (Required) Specify the password that you want to use as a seed to create the encryption algorithm. To increase the security of the secrets, ensure that you change the default password to a unique alphanumeric value.

   Preferred Encryption Method: Specify the preferred encryption method. Select the method that complies with your security model:

   - **Password Based Encryption With MD5 and DES:** MD5 is an algorithm that is used to verify data integrity. Data Encryption Standard (DES) is a widely used method of data encryption that uses a private key.
   - **DES:** Data Encryption Standard (DES) is a widely used method of data encryption that uses a private key. Like other private key cryptographic methods, both the sender and the receiver must know and use the same private key.
   - **Triple DES:** A variant of DES in which data is encrypted three times with standard DES by using two different keys.

5b Specify where to store secret data. (For more information about setting up a user store for secret store, see “Configuring a User Store for Secrets” on page 277.)

- To have the secrets stored in the configuration database, do not configure the list in the **Extended Schema User Store References** section. You only need to configure the fields in Step 5a.
- To store the secrets in your LDAP user store, click **New** in **Extended Schema User Store References** and configure the following fields:

  User Store: Select a user store where secret data is stored.

  Attribute Name: Specify the LDAP attribute of the User object that can be used to store the secrets. When a user authenticates by using the user store specified here, the secret data is stored in an XML document of the specified attribute of the user object. This attribute must be a single-valued case ignore string that you have defined and assigned to the user object in the schema.

  NOTE: Do not use this LDAP attribute in Policy configuration as shared secrets. Instead you create the shared secrets attributes. The Shared secret attributes are populated in the configured LDAP attribute, and are used by policy for mapping. For more information about how to create shared secret, see Chapter , “Form Fill Policies,” on page 711.

- To use Novell SecretStore to remotely store secrets, click **New** under **Novell Secret Store User Store References.**

  Click the user store that you have configured for SecretStore.

  Secure LDAP must be enabled between the user store and the Identity Server to add this user store reference.

5c Click **OK** twice.

6 On the Identity Server page, update the Identity Server.

Customizing Attribute Names

You can change the display names of the attributes for the Credential, Custom, Employee, and Personal profiles. The customized names are displayed on the My Profile page in the User Portal. The users see the custom names applicable to their language. Custom Attributes are displayed on the My Profile page in the User Portal in place of the corresponding English attribute name when the language in the drop-down list is the accepted language of the browser.

1 In the Administration Console, click **Devices > Identity Servers** > *Edit* **> Liberty > Web Service Provider > [Profile] > Custom Attribute Names**.

2 Click the data item name to view the customized attribute names.

3 Click **New** to create a new custom name.

4 Type the name and select a language.

5 Click **OK**.

6 On the Custom Attribute Names page, click **OK**.

7 On the Web Service Provider page, click **OK**.

8 Update the Identity Server configuration on the Servers page.

Configuring the Web Service Consumer

The Web service consumer is the component within the identity provider that requests attributes from Web service providers. The identity provider and Web services consumer cooperate to redirect the user or resource owner to the identity provider, allowing interaction. You can configure an interaction service, which allows the identity provider to pose simple questions to a user. This service can be offered by trusted Web services consumers, or by a dedicated interaction service provider that has a reliable means of communication with the users.

1 In the Administration Console, click **Devices > Identity Servers** > *Edit* **> Liberty > Web Service Consumers**

The following general settings configure time limits and processing speed:

Protocol Timeout (seconds): Limits the time the transport protocol allows.

Provider Timeout (seconds): Limits the request processing at the Web service provider. This value must always be equal to or greater than the **Protocol Timeout** value.

Attribute Cache Enabled: A subsystem of the Web service consumer that caches attribute data that the Web service consumer requests. For example, if the Web service consumer has already requested a first name attribute from a Web service provider, the Web service consumer does not need to request the attribute again. This setting improves performance when enabled. However, you can disable this option to increase system memory.

2 Specify how and when the identity provider interacts with the user:

Always Allow Interaction: Allows interaction to take place between users and service providers.

Never Allow Interaction: Never allows interaction between users and service providers.

Always Allow Interaction for Permissions, Never for Data: Allows interaction for permissions, never for data.

Maximum Allowed Interaction Time: Specifies the allowed time (in seconds).

3 To specify the allowable methods that a Web service provider can use for user interaction, click one of the following options:

Redirect to a User Interaction Service: Allows the Web service consumer to redirect the user agent to the Web service provider to ask questions. After the Web service provider has obtained the information it needs, it can redirect the user back to the Web service consumer.

Call a Trusted User Interaction Service: Allows the Web service provider to trust the Web service consumer to act as proxy for the resource owner.

4 Under **Security Settings**, fill in the following fields:

WSS Security Token Type: Instructs the Web service consumer/requestor how to place the token in the security header as outlined in the Liberty ID-WSF Security Mechanisms.

Signature Algorithm: The signature algorithm to use for signing the payload.

5 Click **OK**, then update the Identity Server configuration as prompted.

Mapping LDAP and Liberty Attributes

You can create an LDAP attribute map or edit an existing one. To create an attribute map, you specify how single-value and multi-value data items map to single-value and multi-value LDAP attributes. A single-value attribute can contain no more than one value, and a multi-value attribute can contain more than one. An example of a single-value attribute might be a person's gender, and an example of a multi-value attribute might be a person's various e-mail addresses, phone numbers, or titles.

1. In the Administration Console, click **Devices**> **Identity Servers** > **Edit** > **Liberty** >**LDAP Attribute Mapping**.
2. Select one of the following actions:

   New: Allows you create an LDAP attribute mapping. Select from the following types:

   - **One to One:** Maps a single Liberty attribute to a single LDAP attribute. See "Configuring One-to-One Attribute Maps" on page 478.
   - **Employee Type:** Maps the Employee Type attribute to an LDAP attribute, then maps the possible Liberty values to LDAP values. See "Configuring Employee Type Attribute Maps" on page 482.
   - **Employee Status:** Maps the Employee Status attribute to an LDAP attribute, then maps the possible Liberty values to LDAP values. See "Configuring Employee Status Attribute Maps" on page 482.
   - **Postal Address:** Maps the Postal Address attribute to either multiple LDAP attributes or a delimited LDAP attribute. See "Configuring Postal Address Attribute Maps" on page 483.
   - **Contact Method:** Maps the Contact Method attribute to multiple LDAP attributes. See "Configuring Contact Method Attribute Maps" on page 484.
   - **Gender:** Maps the Gender attribute to an LDAP attribute, then maps the possible Liberty values to LDAP values. See "Configuring Gender Attribute Maps" on page 485.
   - **Marital Status:** Maps the Marital Status attribute to an LDAP attribute, then maps the possible Liberty values to LDAP values. See "Configuring Marital Status Attribute Maps" on page 485.

   Delete: Deletes the selected mapping.

   Enable: Enables the selected mapping.

   Disable: Disables the selected mapping. When the mapping is disabled, the server does not load the definition. However, the definition is not deleted.
3. Click **OK**, then update the Identity Server.

Configuring One-to-One Attribute Maps

A one-to-one map enables you to map single-value and multiple-value LDAP attribute names to standard Liberty attributes. A default one-to-one attribute map is provided with Access Manager, but you can also define your own.

An example of a one-to-one attribute map might be the single-valued Liberty attribute Common Name (CommonName) used by the Personal Profile that is mapped to the LDAP attribute givenName. You can further configure the various Liberty values to map to any LDAP attribute names that you use.

1. In the Administration Console, click **Devices > Identity Servers > Edit > Liberty > LDAP Attribute Mapping > New > One to One**.
2. Configure the following fields:

   Type: Displays the type of mapping you are modifying or creating:

   Name: The name you want to give the map.

Description: A description of the map.

Access Rights: A drop-down menu that provides the broadest control for the page. If you set this to **Read/Write**, you can specify rights for individual data items.

For user provisioning to succeed, you must select **Read/Write** from the **Access Rights** drop-down menu for any maps that use an attribute during user provisioning.

User Stores: The user store that a map applies to. If a user logs into a user store that is not in the map's user store list, that map is not used to read or write attributes for that user.

3 Use the following guidelines to configure the map:

- Mapping Personal Profile Single-Value Data Items to LDAP Attributes
- Mapping Personal Profile Multiple-Value Data Items to LDAP Attributes
- Mapping Employee Profile Single-Value Data Items to LDAP Attributes
- Mapping Employee Profile Multiple-Value Data Items to LDAP Attributes
- Mapping Custom Profile Single-Value Data Items to LDAP Attributes
- Mapping Custom Profile Multiple-Value Data Items to LDAP Attributes

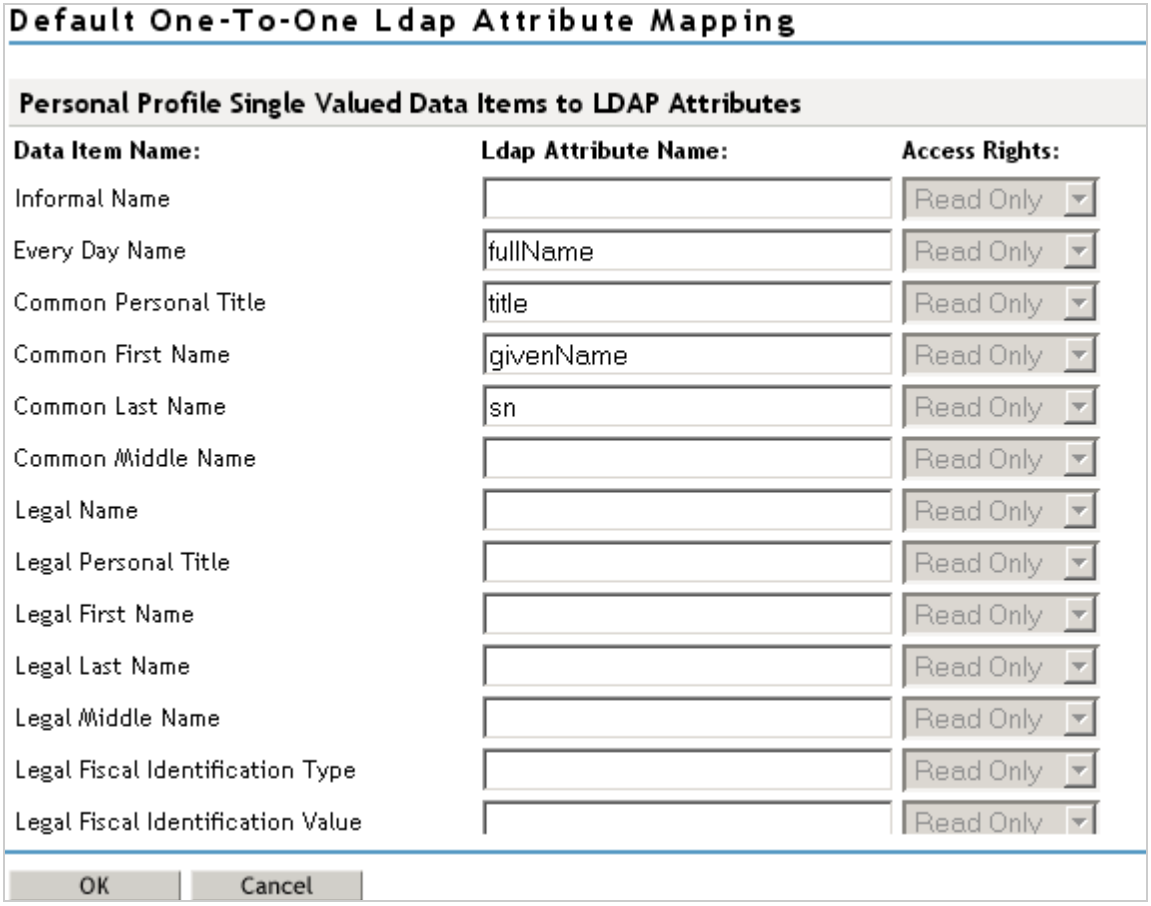
4 After you create the mapping, click **Finish**.

5 On the LDAP Attribute Mapping page, click **OK**.

6 Update the Identity Server.

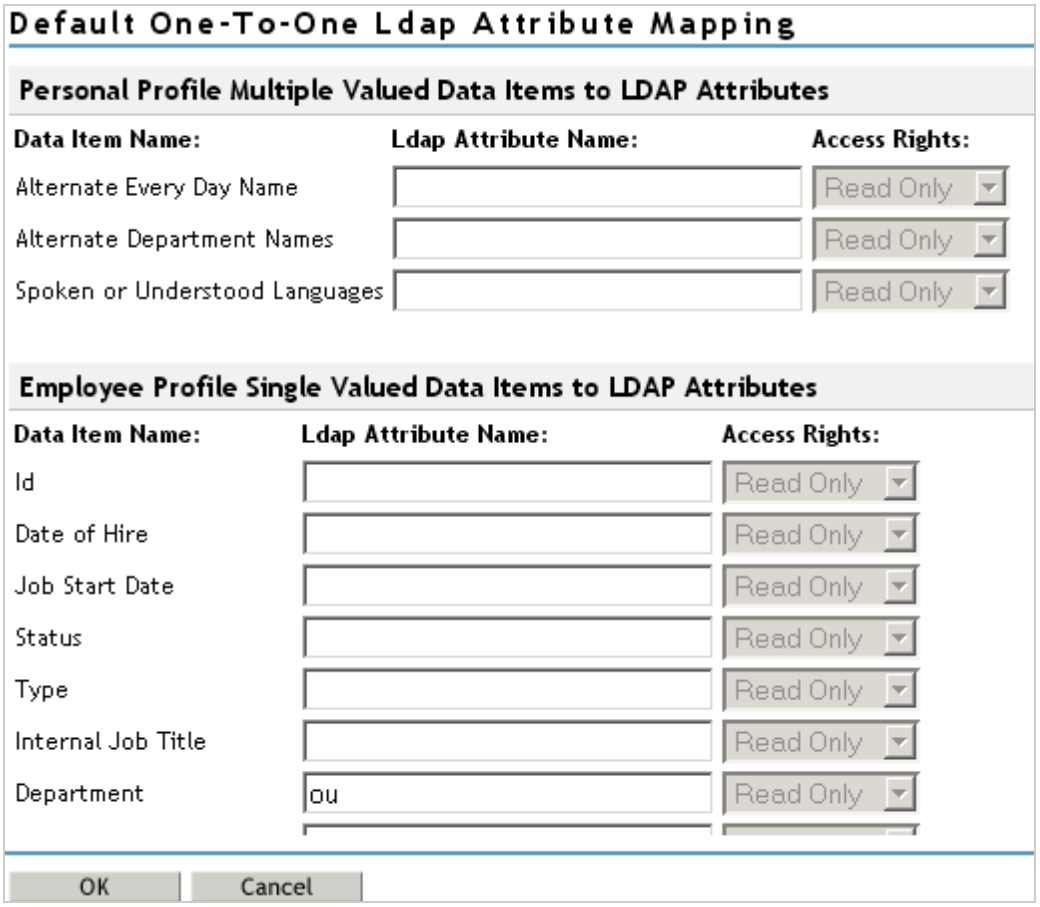
Mapping Personal Profile Single-Value Data Items to LDAP Attributes

The data items displayed are single-value Liberty Personal Profile attributes that you can map to the single-valued LDAP attributes that you have defined for your directory.



Mapping Personal Profile Multiple-Value Data Items to LDAP Attributes

Use the fields on this page to map multiple-value attributes from the Liberty Personal Profile to the multiple-value LDAP attributes you have defined for your directory. For example, you can map the Liberty attribute Alternate Every Day Name (AltCN) to the LDAP attribute you have defined for this purpose in your directory.



Mapping Employee Profile Single-Value Data Items to LDAP Attributes

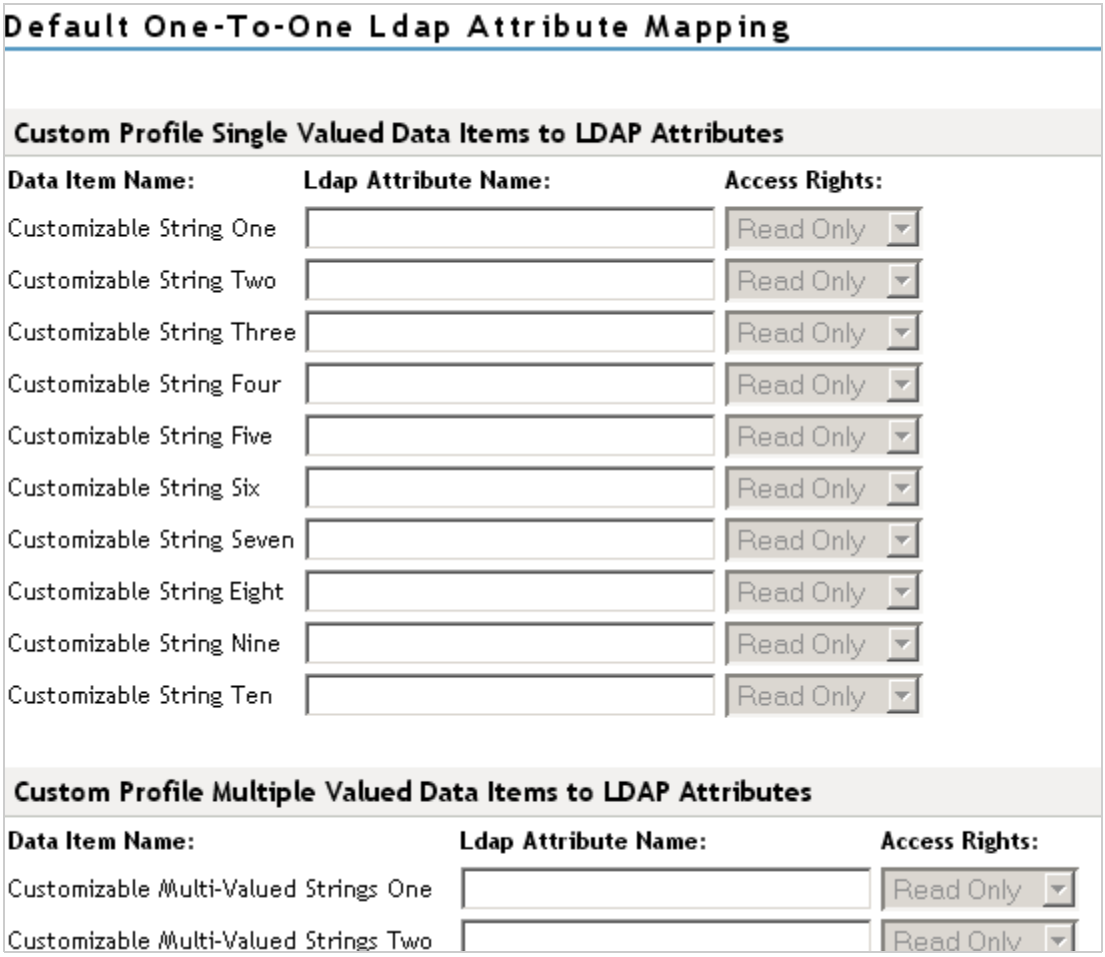
Map the Liberty Employee Profile single-value attributes to the LDAP attributes you have defined in your directory for entries such as ID, Date of Hire, Job Start Date, Department, and so on.

Mapping Employee Profile Multiple-Value Data Items to LDAP Attributes

Map the Liberty Employee Profile multiple-value attributes to the LDAP attributes you have defined in your directory.

Mapping Custom Profile Single-Value Data Items to LDAP Attributes

Map custom Liberty profile single-value attributes to LDAP attributes you have defined in your directory. These attributes are customizable strings associated with the Custom Profile.



Customizable String (1 - 10): The Custom Profile allows custom single-value and multiple-value attributes to be defined without using the Data Model Extension XML to extend a service's schema. To use a customizable attribute, navigate to the **Custom Attribute Names** tab on the Custom Profile Details page (see "Customizing Attribute Names" on page 476). Use the page to customize the name of any of the predefined single-value or multiple-value customizable attributes in the Custom Profile. After you customize a name, you can use that attribute in the same way you use any other profile attribute.

Mapping Custom Profile Multiple-Value Data Items to LDAP Attributes

Customizable Multi-Valued Strings (1 - 5): Similar to customizable strings for single-value attributes, except these attributes can have multiple values. Use this list of fields to map directory attributes that can have multiple values to multiple-value strings from the Custom Profile.

Configuring Employee Type Attribute Maps

You can map the LDAP attribute name and values to the Liberty profile values for Employee Type. This is an Employee Profile attribute. Examples of Liberty values appended to this attribute include Contractor Part Time, Contractor Full Time, Full Time Regular, and so on.

1. In the Administration Console, click **Devices > Identity Servers > Edit > Liberty > LDAP Attribute Mapping > New > Employee Type**.
2. Configure the following fields:

   Name: The name you want to give the map.

   Description: A description of the map.

   Access Rights: A drop-down menu that provide the broadest control for the page. If you set this to **Read/Write**, you can specify rights for individual data items.

   For user provisioning to succeed, you must select **Read/Write** from the **Access Rights** drop-down menu for any maps that use an attribute during user provisioning.

   User Stores: The user store that a map applies to. If a user logs into a user store that is not in the map's user store list, that map is not used to read or write attributes for that user.
3. In the **LDAP Attribute Name** field, type the LDAP attribute name that you want to map to the Liberty Employee Type attribute.
4. In the **LDAP Attribute Value** fields, type the predefined LDAP attribute values that you want to map to the **Liberty Employee Type** values.

   These are the values that you want to store in the LDAP attribute for each given Liberty attribute value. The LDAP attribute map then maps the actual Liberty URI value, back and forth, to this supplied value.
5. Click **Finish**.
6. On the LDAP Attribute Mapping page, click **OK**.
7. Update the Identity Server.

Configuring Employee Status Attribute Maps

You can map the LDAP attribute name and values to the Liberty profile values for Employee Status. This is an Employee Profile attribute. Examples of the values appended to this Liberty attribute include Active, Trial, Retired, Terminated, and so on.

1. In the Administration Console, click **Devices > Identity Servers > Edit > Liberty > LDAP Attribute Mapping > New > Employee Status**.
2. Configure the following fields:

   Name: The name you want to give the map.

   Description: A description of the map.

   Access Rights: A drop-down menu that provide the broadest control for the page. If you set this to **Read/Write**, you can specify rights for individual data items.

   For user provisioning to succeed, you must select **Read/Write** from the **Access Rights** drop-down menu for any maps that use an attribute during user provisioning.

   User Stores: The user store that a map applies to. If a user logs into a user store that is not in the map's user store list, that map is not used to read or write attributes for that user.
3. In the **LDAP Attribute Name** field, type the LDAP attribute name that you want to map to the **Liberty Employee Status** element.

4 In the **LDAP Attribute Value** fields, type the predefined LDAP attribute values that you want to map to the **Liberty Employee Status** values.

These are the values that you want to store in the LDAP attribute for each given Liberty attribute value. The LDAP attribute map then maps the actual Liberty URI value, back and forth, to this supplied value.

5 Click **Finish**.

6 On the LDAP Attribute Mapping page, click **OK**.

7 Update the Identity Server.

Configuring Postal Address Attribute Maps

You can map the LDAP attribute name and values to the Liberty profile values for Postal Address. The PostalAddress element refers to the local address, including street or block with a house number, and so on. This is a Personal Profile attribute.

1 In the Administration Console, click **Devices > Identity Servers > Edit > Liberty > LDAP Attribute Mapping > New > Postal Address**.

2 Configure the following fields:

Name: The name you want to give the map.

Description: A description of the map.

Access Rights: A drop-down menu that provide the broadest control for the page. If you set this to **Read/Write**, you can specify rights for individual data items.

For user provisioning to succeed, you must select **Read/Write** from the **Access Rights** drop-down menu for any maps that use an attribute during user provisioning.

User Stores: The user store that a map applies to. If a user logs into a user store that is not in the map's user store list, that map is not used to read or write attributes for that user.

3 In the **Mode** drop-down menu, select either **Multiple LDAP Attributes** or **Single Delimited LDAP Attributes**.

Multiple LDAP Attributes: Allows you to map multiple LDAP attributes to multiple Liberty Postal Address elements. When you select this option, the following Liberty Postal Address elements are displayed under the **Postal Address to LDAP Attributes** group. Type the LDAP attributes that you want to map to the Liberty elements.

- Postal Address
- Postal Code
- City
- State
- Country

Single Delimited LDAP Attributes: Allows you to specify one LDAP attribute that is used to hold multiple elements of a Liberty Postal Address in a single delimited value. When you select this option, the page displays the following fields:

- **Delimited LDAP Attribute Name:** The delimited LDAP attribute name you have defined for the LDAP postal address that you want to map to the Liberty Postal Address attribute.
- **Delimiter:** The character to use to delimit single-value entries. A $ sign is the default delimiter.

4 (Single Delimited LDAP Attributes mode) Under **One-Based Field Position in Delimited LDAP Attribute**, specify the order in which the information is contained in the string. Select 1 for the value that comes first in the string, 2 for the value that follows the first delimiter, etc.

5 (Multiple LDAP Attributes mode) Under **Postal Address Template Data**, fill in the following options:

Nickname: (Required) A Liberty element name used to identify the Postal Address object.

Contact Method Type: Select the contact method type, such as **Domicile**, **Work**, **Emergency**, and so on.

6 Click **Finish**.

7 On the LDAP Attribute Mapping page, click **OK**.

8 Update the Identity Server.

Configuring Contact Method Attribute Maps

You can map the LDAP attribute you have defined for contact methods to the Liberty attribute Contact Method (MsgContact).

1 In the Administration Console, click **Devices > Identity Servers > Edit > Liberty > LDAP Attribute Mapping > New > Contact Method**.

2 Configure the following fields:

Name: The name you want to give the map.

Description: A description of the map.

Access Rights: A drop-down menu that provide the broadest control for the page. If you set this to **Read/Write**, you can specify rights for individual data items.

For user provisioning to succeed, you must select **Read/Write** from the **Access Rights** drop-down menu for any maps that use an attribute during user provisioning.

User Stores: The user store that a map applies to. If a user logs into a user store that is not in the map's user store list, that map is not used to read or write attributes for that user.

3 Under **Contact Method to LDAP Attributes**, fill in the following fields to map to the Liberty Contact Method attribute:

Provider LDAP Attribute: Maps to the Liberty attribute MsgProvider, which is the service provider or domain that provides the messaging service.

Account LDAP Attribute: Maps to the Liberty attribute MsgAccount, which is the account or address information within the messaging provider.

SubAccount LDAP Attribute: Maps to the Liberty MsgSubaccount, which is the subaccount within a messaging account, such as the voice mail box associated with a phone number.

4 Under **Contact Method Template Data**, specify the settings for the following Liberty attribute values:

Nickname: Maps to the Liberty attribute Nick, which is an informal name for the contact.

Type: Maps to the Liberty attribute MsgType (such as Mobile, Personal, or Work).

Method: Maps to the Liberty MsgMethod (such as Voice, Fax, or E-mail).

Technology: Maps to the Liberty attribute MsgTechnology (such as Pager, VOIP, and so on).

5 Click **Finish**.

6 On the LDAP Attribute Mapping page, click **OK**.

7 Update the Identity Server.

Configuring Gender Attribute Maps

You can map the LDAP attribute name and values to the Liberty profile values for the Gender attribute. You can use gender to differentiate between people with the same name, especially in countries where national ID numbers cannot be collected. This is a Personal Profile attribute.

1. In the Administration Console, click **Devices > Identity Servers > Edit > Liberty > LDAP Attribute Mapping > New > Gender**.
2. Configure the following fields:

   Name: The name you want to give the map.

   Description: A description of the map.

   Access Rights: A drop-down menu that provide the broadest control for the page. If you set this to **Read/Write**, you can specify rights for individual data items.

   For user provisioning to succeed, you must select **Read/Write** from the **Access Rights** drop-down menu for any maps that use an attribute during user provisioning.

   User Stores: The user store that a map applies to. If a user logs into a user store that is not in the map's user store list, that map is not used to read or write attributes for that user.
3. In the **LDAP Attribute Name** field, type the LDAP attribute name that you want to map to the Liberty element Gender.
4. In the **LDAP Attribute Value** fields, type the predefined LDAP attribute values that you want to map to the Gender values.

   These are the values that you want to store in the LDAP attribute for each given Liberty attribute value. The LDAP attribute map then maps the actual Liberty URI value, back and forth, to this supplied value.
5. Click **Finish**.
6. On the LDAP Attribute Mapping page, click **OK**.
7. Update the Identity Server.

Configuring Marital Status Attribute Maps

You can map the LDAP marital status attribute to the Liberty attribute. The Liberty Marital Status (MaritalStatus) element includes appended values such as single, married, divorced, and so on. For example, `urn:liberty:id-sis-pp:maritalstatus:single`. This is a Personal Profile attribute.

1. In the Administration Console, click **Devices > Identity Servers > Edit > Liberty > LDAP Attribute Mapping > New > Marital Status**.
2. Configure the following fields:

   Name: The name you want to give the map.

   Description: A description of the map.

   Access Rights: A drop-down menu that provide the broadest control for the page. If you set this to **Read/Write**, you can specify rights for individual data items.

   For user provisioning to succeed, you must select **Read/Write** from the **Access Rights** drop-down menu for any maps that use an attribute during user provisioning.

   User Stores: The user store that a map applies to. If a user logs into a user store that is not in the map's user store list, that map is not used to read or write attributes for that user.
3. In the **LDAP Attribute Name** field, type the LDAP attribute name that you want to map to the Liberty element Marital Status (MaritalStatus).

4 In the **LDAP Attribute Value** fields, type the predefined LDAP attribute values that you want to map to the MaritalStatus values.

   These are the values that you want to store in the LDAP attribute for each given Liberty attribute value. The LDAP attribute map then maps the actual Liberty URI value, back and forth, to this supplied value.

5 Click **Finish**.

6 On the LDAP Attribute Mapping page, click **OK**.

7 Update the Identity Server.

Configuring WS Federation

The first two topics in this section describe two different methods for setting up federation with a SharePoint server. The next sections describe how you can manage and modify WS Federation providers and configure Security Token Service (STS). STS is used to process authentication requests received at the Identity Server for the WS Federation protocol.

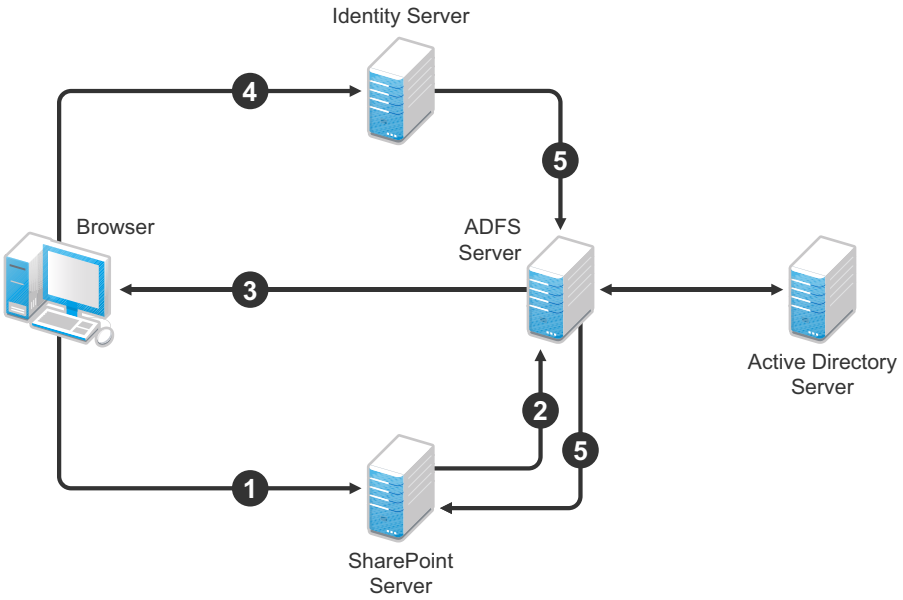
NOTE: WS Federation may fail for .NET applications. For more information about the probable issue, refer TID 7015929.

- "Using the Identity Server as an Identity Provider for ADFS" on page 486
- "Using the ADFS Server as an Identity Provider for an Access Manager Protected Resource" on page 496
- "Managing WS Federation Providers" on page 502
- "Modifying a WS Federation Identity Provider" on page 503
- "Modifying a WS Federation Service Provider" on page 507
- "Configuring STS Attribute Sets" on page 510
- "Configuring STS Authentication Methods" on page 510
- "Configuring STS Authentication Request" on page 510

Using the Identity Server as an Identity Provider for ADFS

The Identity Server can provide authentication for resources protected by an Active Directory Federation Services (ADFS) server. This allows the Identity Server to provide single sign-on to Access Manager resources and ADFS resources, such as a SharePoint server. Figure 5-24 illustrates this configuration.

Figure 5-24 *Accessing SharePoint Resources with an Identity Server*



In this scenario, the following exchanges occur:

1. The user requests access to a SharePoint server protected by the ADFS server.
2. The resource sends an authentication request to the ADFS server.
3. The ADFS server, which has been configured to use the Identity Server as an identity provider, gives the user the option of logging in to the Identity Server.
4. The user logs in to the Identity Server and is provided a token that is sent to the ADFS server and satisfies the request of the resource.
5. The user is allowed to access the resource.

The following section describe how to configure your servers for this scenario:

- “Configuring the Identity Server” on page 487
- “Configuring the ADFS Server” on page 492
- “Logging In” on page 495
- “Troubleshooting” on page 495

Configuring the Identity Server

- “Prerequisites” on page 488
- “Creating a New Authentication Contract” on page 488
- “Setting the WS-Fed Contract to Be the Default Contract” on page 489
- “Enabling the WS Federation Protocol” on page 489
- “Creating an Attribute Set for WS Federation” on page 489
- “Enabling the Attribute Set” on page 490
- “Creating a WS Federation Service Provider” on page 490
- “Configuring the Name Identifier Format” on page 491

- “Setting Up Roles for ClaimApp and TokenApp Claims” on page 491
- “Importing the ADFS Signing Certificate into the NIDP-Truststore” on page 492

Prerequisites

- You have set up the Active Directory Federation Services, Active Directory, and SharePoint servers and the client as described in the ADFS guide from Microsoft. See the “Step-by-Step Guide for Active Directory Federation Services”.
- You have set up the NetIQ Access Manager 4.0 system with a site configuration that is using SSL in the Identity Server's base URL. See Chapter 15, “Enabling SSL Communication,” on page 819.

Creating a New Authentication Contract

The Microsoft ADFS server rejects the contract URI names of the default Access Manager contracts, which have a URI format of `secure/name/password/uri`. The ADFS server expects the URI to look like a URL.

We suggest that you use the following format for the URI of all contracts that you want to use with the ADFS server:

```
<baseurl>/name/password/uri
```

If the DNS name of your Identity Server is `idp-50.amlab.net`, the URI would have the following format:

```
https://idp-50.amlab.net:8443/nidp/name/password/uri
```

This URL does not resolve to anything because the Identity Server interprets it as a contract URI and not a URL.

To create a new authentication contract:

1. In the Administration Console, click **Devices** > **Identity Servers** > **Edit** > **Local** > **Contracts**.
2. Click **New**, then fill in the following fields:

   Display name: Specify a name, for example WS-Fed Contract.

   URI: Specify a URI, for example `https://idp-50.amlab.net:8443/nidp/name/password/uri`.

   Satisfiable by External Provider: Enable this option. The ADFS server needs to satisfy this contract.
3. Move **Name/Password – Form** to the **Methods** list.
4. Click **Next**, then fill in the following fields:

   ID: Leave this field blank. You only need to supply a value when you want a reference that you can use externally.

   Text: Specify a description that is available to the user when the user mouses over the card.

   Image: Select an image, such as **Form Auth Username Password**. This is the default image for the Name/Password - Form contract.

   Show Card: Enable this option so that the card can be presented to the user as a login option.
5. Click **Finish**.
6. Continue with “Setting the WS-Fed Contract to Be the Default Contract” on page 489.

Setting the WS-Fed Contract to Be the Default Contract

It is not possible to specify the contract to request from the ADFS service provider to the Identity Server. You must either set the contract for WS-Fed to be the default, or have your users remember to click that contract every time.

1. On the Local page of the Identity Server, click **Defaults**.
2. For the **Authentication Contract** option, select the WS-Fed Contract.
3. Click **Apply**.
4. Continue with "Enabling the WS Federation Protocol" on page 489.

Enabling the WS Federation Protocol

Access Manager ships with only SAML 1.1, Liberty, and SAML 2.0 enabled by default. To use the WS Federation protocol, you must enable it on the Identity Server.

1. Click the **General** tab.
2. In the **Enabled Protocols** section, select WS Federation.
3. Click **OK**.
4. Update the Identity Server.
5. Continue with "Creating an Attribute Set for WS Federation" on page 489.

Creating an Attribute Set for WS Federation

The WS Federation namespace is `http://schemas.xmlsoap.org/claims`. With WS Federation, you need to decide which attributes you want to share during authentication. This scenario uses the LDAP mail attribute and the All Roles attribute.

1. On the Identity Servers page, click **Shared Settings**.
2. To create a new attribute set, click **New**, then fill in the following fields:

   Set Name: Specify a name that identifies the purpose of the set, for example, wsfed_attributes.

   Select set to use as template: Select **None**.
3. Click **Next**.
4. To add a mapping for the mail attribute:
   - **4a** Click **New**.
   - **4b** Fill in the following fields:

     Local attribute: Select **LDAP Attribute:mail [LDAP Attribute Profile]**.

     Remote attribute: Specify **emailAddress**. This is the attribute that this scenario uses for user identification.

     Remote namespace: Select the radio button by the text box, then specify the following namespace:

     ```
     http://schemas.xmlsoap.org/claims
     ```
   - **4c** Click **OK**.
5. To add a mapping for the All Roles attribute:
   - **5a** Click **New**.
   - **5b** Fill in the following fields:

     Local attribute: Select **All Roles**.

Remote attribute: Specify **group**. This is the name of the attribute that is used to share roles.

Remote namespace: Select the radio button by the text box, then specify the following namespace:

```
http://schemas.xmlsoap.org/claims
```

5c Click **OK**.

6 Click **Finish**.

7 Continue with "Enabling the Attribute Set" on page 490.

Enabling the Attribute Set

Because the WS Federation protocol uses STS, you must enable the attribute set for STS to use it in an WS Federation relationship.

1. On the Identity Servers page, click **Servers** > **Edit** > **WS Federation** > **STS Attribute Sets**.
2. Move the WS Federation attribute set to the **Attribute sets** list.
3. Select the WS Federation attribute set and use the up-arrow to make it first in the **Attribute set** list.
4. Click **OK**, then update the Identity Server.

Creating a WS Federation Service Provider

To establish a trusted relationship with the ADFS server, you need to set up the Trey Research site as a service provider. The trusted relationship allows the service provider to trust the Identity Server for user authentication credentials.

Trey Research is the default name for the ADFS resource server. If you have used another name, substitute it when following these instructions. To create a service provider, you need to know the following about the ADFS resource server.

Table 5-16 *ADFS Resource Server Information*

What You Need to Know	Default Value and Description
Provider ID	**Default Value:** `urn:federation:treyresearch` This is the value that the ADFS server provides to the Identity Server in the realm parameter of the query string. This value is specified in the Properties of the Trust Policy page on the ADFS server. The parameter label is **Federation Service URI**.
Sign-on URL	**Default Value:** `https://adfsresource.treyresearch.net/adfs/ls/` This is the value that the identity provider redirects the user to after login. Although it is listed as optional, and is optional between two NetIQ Identity Servers, the ADFS server does not send this value to the identity provider. It is required when setting up a trusted relationship between an ADFS server and a NetIQ Identity Server. This URL is listed in the Properties of the Trust Policy page on the ADFS server. The parameter label is **Federation Services endpoint URL**.
Logout URL	**Default Value:** `https://adfsresource.treyresearch.net/adfs/ls/` This parameter is optional. If it is specified, the user is logged out of the ADFS server and the Identity Server.

What You Need to Know	Default Value and Description
Signing Certificate	This is the certificate that the ADFS server uses for signing. You need to export it from the ADFS server. It can be retrieved from the properties of the **Trust Policy** on the ADFS Server on the **Verification Certificates** tab.This certificate is a self-signed certificate that you generated when following the Active Directory step-by-step guide.

To create a service provider configuration:

1. On the Identity Servers page, click **Edit** > **WS Federation**.
2. Click **New** > **Service Provider**, then fill in the following fields:

   Name: Specify a name that identifies the service provider, such as `TreyResearch`.

   Provider ID: Specify the provider ID of the ADFS server. The default value is `urn:federation:treyresearch`.

   Sign-on URL: Specify the URL that the user is redirected to after login. The default value is `https://adfsresource.treyresearch.net/adfs/ls/`.

   Logout URL: (Optional) Specify the URL that the user can use for logging out. The default value is `https://adfsresource.treyresearch.net/adfs/ls`.

   Service Provider: Specify the path to the signing certificate of the ADFS server.
3. Click **Next**, confirm the certificate, then click **Finish**.
4. Continue with "Configuring the Name Identifier Format" on page 491.

Configuring the Name Identifier Format

The Unspecified Name Identifier format is the default for a newly created WS Federation service provider, but this name identifier format does not work with the ADFS federation server. Additionally, some Group Claims (Adatum ClaimApp Claim and Adatum TokenApp Claim) must be satisfied in order to gain access to the SharePoint server.

1. On the WS Federation page, click the name of the TreyResearch service provider.
2. Click **Attributes**, then fill in the following fields:

   Attribute set: Select the WS Federation attribute set you created.

   Send with authentication: Move the All Roles attribute to the **Send with authentication** list.
3. Click **Apply**, then click **Authentication Response**.
4. Select **E-mail** for the Name Identifier Format.
5. Select **LDAP Attribute:mail [LDAP Attribute Profile]** as the value for the e-mail identifier.
6. Click **OK** twice, then update the Identity Server.
7. Continue with "Setting Up Roles for ClaimApp and TokenApp Claims" on page 491.

Setting Up Roles for ClaimApp and TokenApp Claims

When users access resources on the ADFS server, they need to have two roles assigned: a ClaimApp role and a TokenApp role. The following steps explain how to create these two roles so that they are assigned to all users that log in to the Identity Server.

1. On the Identity Servers page, click **Edit** > **Roles** > **Manage Policies**.
2. Click **New**, specify a name for the policy, select **Identity Server: Roles**, then click **OK**.
3. On the Rule 1 page, leave Condition Group 1 blank.

With no conditions to match, this rule matches all authenticated users.

4 In the **Actions** section, click **New** > **Activate Role**.

5 In the text box, specify **ClaimApp**.

6 In the **Actions** section, click **New** > **Activate Role**.

7 In the text box, specify **TokenApp**.

8 Click **OK** twice, then click **Apply Changes**.

9 Click **Close**.

10 On the Roles page, select the role policy you just created, then click **Enable**.

11 Click **OK**, then update the Identity Server.

12 Continue with "Importing the ADFS Signing Certificate into the NIDP-Truststore" on page 492.

Importing the ADFS Signing Certificate into the NIDP-Truststore

The NetIQ Identity Provider (NIDP) must have the trusted root of the ADFS signing certificate (or the certificate itself) listed in its Trust Store, as well as specified in the relationship. This is because most ADFS signing certificates are part of a certificate chain, and the certificate that goes into the metadata is not the same as the trusted root of that certificate. However, because the Active Directory step-by-step guide uses self-signed certificates for signing, it is the same certificate in both the Trust Store and in the relationship.

To import the ADFS signing certificate's trusted root (or the certificate itself) into the NIDP-Truststore:

1 On the Identity Servers page, click **Edit** > **Security** > **NIDP Trust Store**.

2 Click **Add**.

3 Next to the **Trusted Root(s)** field, click the **Select Trusted Root(s)** icon.

This adds the trusted root of the ADFS signing certificate to the Trust Store.

4 On the Select Trusted Roots page, select the trusted root or certificate that you want to import, then click **Add Trusted Roots to Trust Stores**.

If there is no trusted root or certificate in the list, click **Import**. This enables you to import a trusted root or certificate.

5 Next to the **Trust store(s)** field, click the **Select Keystore** icon.

6 Select the trust stores where you want to add the trusted root or certificate, then click **OK** twice.

7 Update the Identity Server so that the changes can take effect.

This finishes the configuration that must be done on the Identity Server for the Identity Server to trust the ADFS server. The ADFS server must be configured to trust the Identity Server. Continue with "Configuring the ADFS Server" on page 492.

Configuring the ADFS Server

The following tasks must be completed on the Trey Research server (adfsresouce.treyresearch.net) to establish trust with the NetIQ Identity Server.

- "Enabling E-mail as a Claim Type" on page 493
- "Creating an Account Partners Configuration" on page 493
- "Enabling ClaimApp and TokenApp Claims" on page 494
- "Disabling CRL Checking" on page 494

Enabling E-mail as a Claim Type

There are three types of claims for identity that can be enabled on an ADFS server. They are Common Name, E-mail, and User Principal Name. The ADFS step-by-step guide specifies that you do everything with a User Principal Name, which is an Active Directory convention. Although it could be given an e-mail name that looks the same, it is not. This scenario selects to use E-mail instead of Common Name because E-mail is a more common configuration.

1. From the Administrative Tools, open the Active Directory Federation Services tool.
2. Navigate to the **Organizational Claims** by clicking **Federation Service** > **Trust Policy** > **My Organization**.
3. Verify that E-mail is in this list. If it isn't, move it to the list.
4. Navigate to your Token-based Application and enable e-mail by right-clicking the application, editing the properties, and clicking the **Enabled** box.
5. Navigate to your Claims-aware Application and repeat the process.
6. Continue with "Creating an Account Partners Configuration" on page 493.

Creating an Account Partners Configuration

WS Federation requires a two-way trust relationship. Both the identity provider and the service provider must be configured to trust the other provider. This task sets up the trust between the ADFS server and the Identity Server.

1. In the Active Directory Federation Services console, navigate to the Account Partners by clicking **Federation Services** >**Trust Policy** > **Partner Organizations**.
2. Right-click **Partner Organizations**, then select **New** > **Account Partner**.
3. Supply the following information in the wizard:
   - You do not have an account partner policy file.
   - For the display name, specify the DNS name of the Identity Server.
   - For the **Federation Services URI,** specify the following:

     `https://<DNS_Name>:8443/nidp/wsfed/`

     Replace <*DNS_Name*> with the DNS name of the Identity Server.

     This URI is the base URL of your Identity Server with the addition of `/wsfed/` on the end.
   - For the **Federation Services endpoint URL**, specify the following:

     `https://<DNS_Name>:8443/nidp/wsfed/ep`

     Replace <*DNS_Name*> with the DNS name of the Identity Server.

     This URL is the base URL of your Identify Server with the addition of `/wsfed/ep` at the end.
   - For the verification certificate, import the trusted root of the signing certificate on your Identity Server.

     If you have not changed it, you need the Organizational CA certificate from your Administration Console. This is the trusted root for the test-signing certificate.
   - Select **Federated Web SSO**.

     The Identity Server is outside of any forest, so do not select **Forest Trust**.
   - Select the E-mail claim.

- Add the suffix that you will be using for your e-mail address.

  You need to have the e-mail end in a suffix that the ADFS server is expecting, such as @novell.com, which grants access to any user with that e-mail suffix.

4 Enable this account partner.

5 Finish the wizard.

6 Continue with "Enabling ClaimApp and TokenApp Claims" on page 494.

Enabling ClaimApp and TokenApp Claims

The Active Directory step-by-step guide sets up these roles to be used by the resources. You set them up to be sent in the All Roles attribute from the Identity Server. You must map these roles into the Adatum ClaimApp Claim and the Adatum TokenApp Claim.

1 In the Active Directory Federation Services console, click the account partner that you created for the Identity Server (see "Creating an Account Partners Configuration" on page 493).

2 Right click the account partner, then create a new **Incoming Group Claim Mapping** with the following values:

  Incoming group claim name: Specify **ClaimApp**.

  Organization group claim: Specify **Adatum ClaimApp Claim**.

3 Right-click the account partner, and create another **Incoming Group Claim Mapping** with the following values:

  Incoming group claim name: Specify **TokenApp**.

  Organization group claim: Specify **Adatum TokenApp Claim**.

4 Continue with "Disabling CRL Checking" on page 494.

Disabling CRL Checking

If you are using the Access Manager certificate authority as your trusted root for the signing certificate (test-signing certificate), there is no CRL information in that certificate. However, the ADFS has a mandatory requirement to do CRL checking on any certificate that they receive. For instructions on how to disable this checking, see "Turn CRL checking on or off".

Use the following tips as you follow these instructions.

- Create a file from the script contained at that link called `TpCrlChk.vbs`.
- Exit the Active Directory Federation Services console.

  If you do not exit the console, the console overwrites the changes made by the script file and CRL checking is not turned off.
- Run the command with the following syntax:

  ```
  Cscript TpCrlChk.vbs <location of ADFS>\TrustPolicy.xml "<service URI>" None
  ```

  Replace *<location of ADFS>* with the location of the ADFS `TrustPolicy.xml` file. The default location is `C:\ADFS\TrustPolicy.xml`.

  Replace *<service URI>* with the URI you specified in Step 3 on page 493. If the DNS name of your Identity Server is `idp-50.amlab.net`, replace it with `https://idp-50.amlab.ne:8443/nidp/wsfed/`.

  Your command must look similar to the following:

  ```
  Cscript TpCrlChk.vbs C:\ADFS\TrustPolicy.xml "https://idp-50.amlab.net:8443/nidp/wsfed/" None
  ```

Logging In

1. In a browser on your client machine, enter the URL of the SharePoint server. For example:

   ```
   https://adfsweb.treyresearch.net/default.aspx
   ```

2. Select the IDP from the drop-down list of **home realm**, then submit the request.

   If you are not prompted for the realm, clear all cookies in the browser and try again.

3. Log in with a user at the NetIQ Identity Provider
4. Verify that you can access the SharePoint server.If you only see a page that says `Server Error in '/adfs' Application`, see "Turning On Logging on the ADFS Server" on page 495 and follow the instructions in "Common Errors" on page 495.

Troubleshooting

- "Turning On Logging on the ADFS Server" on page 495
- "Common Errors" on page 495

Turning On Logging on the ADFS Server

If you see the message `Server Error in '/adfs' Application` displayed in the client's browser, the best place to look for the cause is in the ADFS log file.

To turn on this log file:

1. In the Active Directory Federation Services console, right-click **Federation Service**, then click **Properties**.
2. Click the **Troubleshooting** tab, then enable everything on the page.
3. Click **OK**, then look for the file that is created in the path listed in the **Log files directory**.
4. Look in that file for reasons that the federation is failing.

   For an explanation of some of the common errors, see "Common Errors" on page 495.

Common Errors

- "[ERROR] SamlViolatesSaml:" on page 495
- "[ERROR] Saml contains an unknown NameIdentifierFormat:" on page 495
- "CRL Errors" on page 496
- "[ERROR] EmailClaim.set_Email:" on page 496

[ERROR] SamlViolatesSaml:

Error parsing AuthenticationMethod: Invalid URI: The format of the URI could not be determined.

Cause: This is because the contract has the wrong format for its URI. The URI must start with `urn:` or `http://`. Change the contract and try again.

[ERROR] Saml contains an unknown NameIdentifierFormat:

Issuer=https://idp-51.amlab.net:8443/nidp/wsfed/; Format=urn:oasis:names:tc:SAML:1.1:nameid-format:unspecified

Cause: The name identifier format is set to unspecified, and it needs to be set to E-mail.

[ERROR] Saml contains an unknown Claim name/namespace:

Issuer=https://idp-51.amlab.net:8443/nidp/wsfed/;
Namespace=urn:oasis:names:tc:SAML:1.0:assertion; Name=emailaddress

Cause: The emailAddress attribute is not in the correct namespace for WSFed.

CRL Errors

- 2008-08-01T19:56:55 [WARNING] VerifyCertChain: Cert chain did not verify - error code was 0x80092012
- 2008-08-01T19:56:55 [ERROR] KeyInfo processing failed because the trusted certificate does not have a a valid certificate chain. Thumbprint = 09667EB26101A98F44034A3EBAAF9A3A09A0F327
- 2008-08-01T19:56:55 [WARNING] Failing signature verification because the KeyInfo section failed to produce a key.
- 2008-08-01T19:56:55 [WARNING] SAML token signature was not valid: AssertionID = idZ0KQH0kfjVK8kmKfv6YaVPglRNo

Cause: The CRL check isn't turned off. See “Disabling CRL Checking” on page 494.

[ERROR] EmailClaim.set_Email:

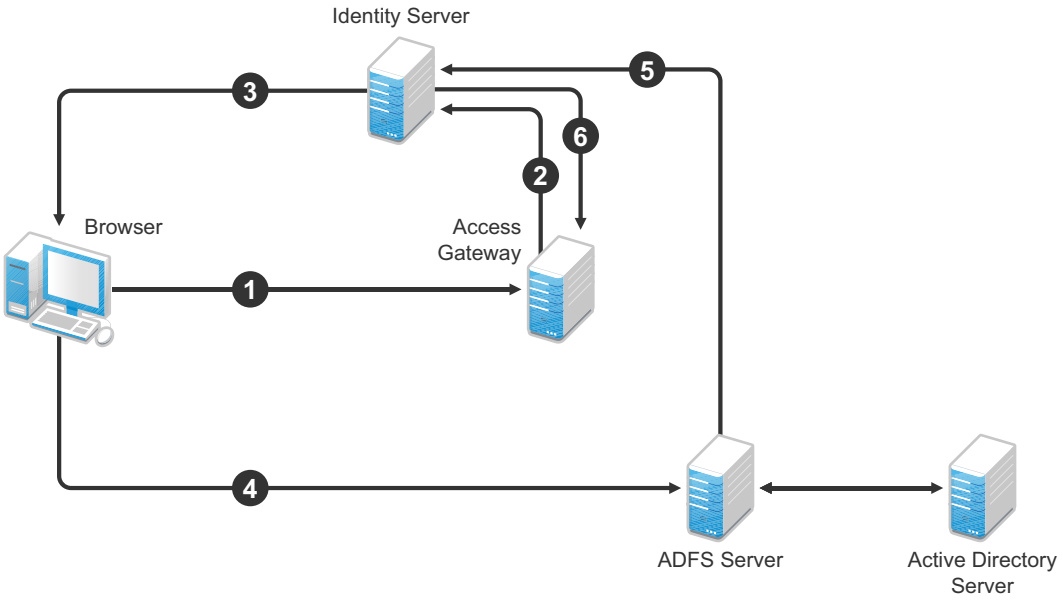
Email 'mPmNXOA8Rv+j16L1iNKn/4HVpfeJ3av1L9c0GQ==' has invalid format

Cause: The drop-down list next to E-mail in the identifier format was not changed from <Not Specified> to a value with a valid e-mail address in it.

Using the ADFS Server as an Identity Provider for an Access Manager Protected Resource

The Active Directory Federation Services server can be configured to provide authentication for a resource protected by Access Manager.

Figure 5-25 *Using an ADFS Server for Access Manager Authentication*



In this scenario, the following exchanges occur:

1. The user requests access to a resource protected by an Access Gateway.
2. The resource sends an authentication request to the NetIQ Identity Server.
3. The Identity Server is configured to trust an Active Directory Federation Services server and gives the user the option of logging in at the Active Directory Federation Services server.
4. The user logs into the Active Directory Federation Services server and is provided a token
5. The token is sent to the Identity Server.
6. The token satisfies the authentication requirements of the resource, so the user is allowed to access the resource.

The following sections describe how to configure this scenario.

- "Configuring the Identity Server as a Service Provider" on page 497
- "Configuring the ADFS Server to Be an Identity Provider" on page 500
- "Logging In" on page 501
- "Additional WS Federation Configuration Options" on page 501

Configuring the Identity Server as a Service Provider

- "Prerequisites" on page 497
- "Enabling the WS Federation Protocol" on page 497
- "Creating a WS Federation Identity Provider" on page 498
- "Modifying the User Identification Specification" on page 499
- "Importing the ADFS Signing Certificate into the NIDP-Truststore" on page 499

Prerequisites

- You have set up the Active Directory Federation Services, Active Directory, and SharePoint servers and the client as described in the ADFS guide from Microsoft. See the "Step-by-Step Guide for Active Directory Federation Services".
- You have set up the NetIQ Access Manager system with a site configuration that is using SSL in the Identity Server's base URL. See Chapter 15, "Enabling SSL Communication," on page 819.
- Enable the Liberty Personal Profile.

  In the Administration Console, click **Identity Servers** > **Edit** > **Liberty** > **Web Service Provider**. Select the **Personal Profile**, then click **Enable** > **Apply**. Update the Identity Server.

Enabling the WS Federation Protocol

Access Manager ships with only SAML 1.1, Liberty, and SAML 2.0 enabled by default. To use the WS Federation protocol, it must be enabled on the Identity Server.

1. In the Administration Console, click **Devices** > **Identity Servers** > **Edit**.
2. In the **Enabled Protocols** section of the General Configuration page, select **WS Federation**.
3. Click **OK**.
4. Update the Identity Server.
5. Continue with "Creating a WS Federation Identity Provider" on page 498.

Creating a WS Federation Identity Provider

To have a trust relationship, you need to set up the Adatum site (adfsaccount.adatum.com) as an identity provider for the Identity Server.

Adatum is the default name for the identity provider. If you have used another name, substitute it when following these instructions. To create an identity provider, you need to know the following information about the Adatum site:

Table 5-17 *Adatum Values*

What You Need to Know	Default Value and Description
Provider ID	**Default Value:** `urn:federation:adatum` The ADFS server provides this value to the service provider in the realm parameter in the assertion. You set this value in the **Properties** of the Trust Policy on the ADFS server. The label is **Federation Service URI**.
Sign-on URL	**Default Value:** `https://adfsaccount.adatum.com/adfs/ls/` The service provider uses this value to redirect the user for login. This URL is listed in the **Properties** of the Trust Policy on the ADFS server. The label is **Federation Services endpoint URL**.
Logout URL	**Default Value:** `https://adfsresource.treyresearch.net/adfs/ls/` The ADFS server makes no distinction between the login and logout URL. Access Manager has separate URLs for login and logout, but from a NetIQ Identity Server to an ADFS server, they are the same.
Signing Certificate	This is the certificate that the ADFS server uses for signing. You need to export it from the ADFS server. It can be retrieved from the properties of the **Trust Policy** on the ADFS Server on the **Verification Certificates** tab.This certificate is a self-signed certificate that you generated when following the step-by-step guide.

To create an identity provider:

1. In the Administration Console, click **Devices** > **Identity Servers** > **Edit** > **WS Federation**.
2. On the WS Federation page, click **New**, select **Identity Provider**, then fill in the following fields:

   Name: Specify a name that identifies the identity provider, such as `Adatum`.

   Provider ID: Specify the federation service URI of the identity provider, for example `urn:federation:adatum`.

   Sign-on URL: Specify the URL for logging in, such as `https://adfsaccount.adatum.com/adfs/ls/`.

   Logout URL: Specify the URL for logging out, such as `https://adfsresource.treyresearch.net/adfs/ls/`

   Identity Provider: Specify the path to the signing certificate of the ADFS server.
3. Confirm the certificate, then click **Next**.
4. For the authentication card, specify the following values:

   ID: Leave this field blank.

   Text: Specify a description that is available to the user when the user mouses over the card.

Image: Select an image, such as **Customizable**, or any other image.

Show Card: Enable this option so that the card can be presented to the user as a login option.

5 Click **Finish**.

6 Continue with "Modifying the User Identification Specification" on page 499.

Modifying the User Identification Specification

The default settings for user identification are set to do nothing. The user can authenticated but the user is not identified as a local user on the system. This is not the scenario we are configuring. We want the user to be identified on the local system. Additionally, we want to specify which contract on the Access Gateway is satisfied with this identification. If a contract is not specified, the Access Gateway resources must be configured to use the **Any Contract** option, which is not a typical configuration.

1 On the WS Federation page, click the name of the Adatum identity provider configuration.

2 Click **User Identification**.

3 For **Satisfies contract**, select **Name/Password – Form**.

4 Select **Allow federation**.

5 For the **User Identification Method**, select **Authenticate**.

6 Click **OK** twice.

7 Update the Identity Provider.

8 Continue with "Importing the ADFS Signing Certificate into the NIDP-Truststore" on page 499.

Importing the ADFS Signing Certificate into the NIDP-Truststore

The Identity Server must have the trusted root of the ADFS signing certificate (or the certificate itself) listed in its trust store, as well as specified in the relationship. This is because most ADFS signing certificates have a chain, and the certificate that goes into the metadata is not the same as the trusted root of that certificate. However, because the Active Directory step-by-step guide uses self-signed certificates for signing, it is the same certificate in both the trust store and in the relationship.

To import the ADFS signing certificate's trusted root (or the certificate itself) into the NIDP-Truststore:

1 On the Identity Servers page, click **Edit** > **Security** > **NIDP Trust Store**.

2 Click **Add**.

3 Next to the **Trusted Root(s)** field, click the **Select Trusted Root(s)** icon.

This adds the trusted root of the ADFS signing certificate to the Trust Store.

4 On the Select Trusted Roots page, select the trusted root or certificate that you want to import, then click **Add Trusted Roots to Trust Stores**.

If there is no trusted root or certificate in the list, click **Import**. This enables you to import a trusted root or certificate.

5 Next to the **Trust store(s)** field, click the **Select Keystore** icon.

6 Select the trust stores where you want to add the trusted root or certificate, then click **OK** twice.

7 Update the Identity Server so that changes can take effect.

This ends the basic configuration that must be done to for the Identity Server to trust the ADFS server as an identity provider. However, the ADFS server needs to be configured to act as an identity server and to trust the Identity Server. Continue with "Configuring the ADFS Server to Be an Identity Provider" on page 500.

Configuring the ADFS Server to Be an Identity Provider

The following tasks describe the minimum configuration required for the ADFS server to act as an identity provider for the Access Manager Identity Server.

- “Enabling a Claim Type for a Resource Partner” on page 500
- “Creating a Resource Partner” on page 500

For additional configuration options, see “Additional WS Federation Configuration Options” on page 501.

Enabling a Claim Type for a Resource Partner

You can enable three types of claims for identity on an ADFS Federation server. They are Common Name, E-mail, and User Principal Name. The ADFS step-by-step guide specifies that you do everything with a User Principal Name, which is an Active Directory convention. Although it could be given an e-mail that looks the same, it is not. This scenario selects to use E-mail instead of Common Name because E-mail is a more common configuration.

1. In the Administrative Tools, open the **Active Directory Federation Services** tool.
2. Navigate to the **Organizational Claims** by clicking **Federation Service** > **Trust Policy** > **My Organization**.
3. Ensure that E-mail is in this list.
4. Navigate to Active Directory by clicking **Federation Services** > **Trust Policy** > **Account Stores**.
5. Enable the **E-mail Organizational Claim**:
   - **5a** Right-click this claim, then select **Properties**.
   - **5b** Click the **Enabled** box.
   - **5c** Add the LDAP mail attribute by clicking **Settings** > **LDAP attribute** and selecting **mail**.

     This is the LDAP attribute in Active Directory where the user’s e-mail address is stored.
   - **5d** Click **OK**.
6. Verify that the user you are going to use for authentication has an E-mail address in the mail attribute.
7. Continue with “Creating a Resource Partner” on page 500.

Creating a Resource Partner

The WS Federation protocol requires a two-way trust. The identity provider must be configured to trust the service provider, and the service provider must be configured to trust the identity provider. You have already set up the service provider to trust the identity provider (see “Creating a WS Federation Identity Provider” on page 498). This section sets up the trust so that the identity provider (the ADFS server) trusts the service provider (the Identity Server).

1. In the Active Directory Federation Services console, access the Resource Partners page by clicking **Federation Services** > **Trust Policy** > **Partner Organizations**.
2. Right-click the **Partner Organizations**, then click **New** > **Resource Partner**.
3. Supply the following information in the wizard:
   - You do not have a resource partner policy file to import.
   - For the display name, specify the DNS name of the Identity Server.
   - For the **Federation Services URI**, enter the following:

     `https://<DNS_Name>:8443/nidp/wsfed/`

Replace <*DNS_Name*> with the name of your Identity Server.

This is the base URL of your Identity Server with the addition of `/wsfed/` at the end.

- For the Federation Services endpoint URL, specify the following:

  `https://<DNS_Name>:8443/nidp/wsfed/spassertion_consumer`

  Replace <*DNS_Name*> with the name of your Identity Server.

  This is the base URL of your Identity Server with the addition of `/wsfed/spassertion_consumer` at the end.

- Select **Federated Web SSO**.

  The Identity Server is outside of any forest, so do not select **Forest Trust**.

- Select the E-mail claim.
- Select the **Pass all E-mail suffixes through unchanged** option.

4 Enable this resource partner.

5 Finish the wizard.

6 To test the configuration, continue with "Logging In" on page 501.

Logging In

1 In a client browser, enter the base URL of your Identity Server.

2 From the list of cards, select the Adatum contract.

3 (Conditional) If you are not joined to the Adatum domain, enter a username and password in the browser pop-up. Use a name and a password that are valid in the Adatum domain.

If you are using the client that is joined to the Adatum domain, the card uses a Kerberos ticket to authenticate to the ADFS identity provider (resource partner).

4 When you are directed back to the Identity Server for Federation User Identification, log in to the Identity Server with a username and password that is valid for the Identity Server (the service provider).

5 Verify that you are authenticated.

6 Close the browser.

7 Log in again.

This time you are granted access without entering credentials at the service provider.

Additional WS Federation Configuration Options

You can enable the sharing of attribute information from the Identity Server to the ADFS server. This involves creating an attribute set and enabling the sending of the attributes at authentication. See "Configuring the Attributes Obtained at Authentication" on page 504.

For other options that can be modified after you have created the trusted identity server configuration, see "Modifying a WS Federation Identity Provider" on page 503.

Managing WS Federation Providers

The WS Federation page allows you to create or edit trusted identity providers and trusted service providers. When you create an identity provider configuration, you are configuring the Identity Server to be a WS Federation resource partner. When you create a service provider configuration, you are configuring the Identity Server to be a WS Federation account partner.

1 In the Administration Console, click **Devices > Identity Servers > Edit > WS Federation**.

2 Select one of the following actions:

New: Launches the Create Trusted Identity Provider Wizard or the Create Trusted Service Provider Wizard, depending on your selection. For more information, see one of the following:

- “Creating an Identity Provider for WS Federation” on page 502
- “Creating a Service Provider for WS Federation” on page 503

Delete: Allows you to delete the selected identity or service provider. This action deletes the definition.

Enable: Enables the selected identity or service provider.

Disable: Disables the selected identity or service provider. When the provider is disabled, the server does not load the definition. However, the definition is not deleted.

Modify: Click the name of a provider. For configuration information, see “Modifying a WS Federation Identity Provider” on page 503 or “Modifying a WS Federation Service Provider” on page 507.

3 Click **OK**, then update the Identity Server.

Creating an Identity Provider for WS Federation

To have a trust relationship, you need to set up the ADFS server as an identity provider for the Identity Server.

1 In the Administration Console, click **Devices** > **Identity Servers** > **Edit** > **WS Federation**.

2 On the WS Federation page, click **New**, select **Identity Provider**, then fill in the following fields:

Name: Specify a name that identifies the identity provider, such as `Adatum`.

Provider ID: Specify the federation service URI of the identity provider, for example `urn:federation:adatum`.

Sign-on URL: Specify the URL for logging in, such as `https://adfsaccount.adatum.com/adfs/ls/`.

Logout URL: Specify the URL for logging out, such as `https://adfsresource.treyresearch.net/adfs/ls/`

Identity Provider: Specify the path to the signing certificate of the ADFS server.

3 Confirm the certificate, then click **Next**.

4 For the authentication card, specify the following values:

ID: Leave this field blank.

Text: Specify a description that is available to the user when the user mouses over the card.

Image: Select an image, such as **Customizable**, or any other image.

Show Card: Enable this option so that the card can be presented to the user as a login option.

5 Click **Finish**.

For information about additional configuration steps required to use this identity provider, see “Using the ADFS Server as an Identity Provider for an Access Manager Protected Resource” on page 496.

Creating a Service Provider for WS Federation

To establish a trusted relationship with the ADFS server, you need to set up the ADFS server as service provider. The trusted relationship allows the service provider to trust the Identity Server for user authentication credentials.

1. In the Administration Console, click **Devices** > **Identity Servers** > **Edit** > **WS Federation**.
2. Click **New** > **Service Provider**, then fill in the following fields:

   Name: Specify a name that identifies the service provider, such as `TreyResearch`.

   Provider ID: Specify the provider ID of the ADFS server. The default value is `urn:federation:treyresearch`.

   Sign-on URL: Specify the URL that the user is redirected to after login. The default value is `https://adfsresource.treyresearch.net/adfs/ls/`.

   Logout URL: (Optional) Specify the URL that the user can use for logging out. The default value is `https://adfsresource.treyresearch.net/adfs/ls`.

   Service Provider: Specify the path to the signing certificate of the ADFS server.
3. Click **Next**, confirm the certificate, then click **Finish**.

For information about additional configuration steps required to use this service provider, see “Using the Identity Server as an Identity Provider for ADFS” on page 486.

Modifying a WS Federation Identity Provider

This section explains how to modify a WS Federation identity provider after it has been created. “Creating an Identity Provider for WS Federation” on page 502 explains the steps required to create an identity provider. You can modify the following configuration details:

- “Renaming the Trusted Provider” on page 503
- “Configuring the Attributes Obtained at Authentication” on page 504
- “Modifying the User Identification Method” on page 504
- “Viewing the WS Identity Provider Metadata” on page 505
- “Editing the WS Identity Provider Metadata” on page 506
- “Modifying the Authentication Card” on page 506
- “Assertion Validity Window” on page 507

Renaming the Trusted Provider

1. In the Administration Console, click **Devices > Identity Servers > Edit > WS Federation > [Provider Name].**
2. In the **Name** field, specify a new name for the trusted provider.
3. Click **OK** twice, then update the Identity Server.

Configuring the Attributes Obtained at Authentication

When the Identity Server creates its request to send to the identity provider, it uses the attributes that you have selected. The request asks the identity provider to provide values for these attributes. You can then use these attributes to create policies, to match user accounts, or if you allow provisioning, to create a user account on the service provider.

To select the attributes:

1. In the Administration Console, click **Devices > Identity Servers > Edit > WS Federation > [Identity Provider] > Attributes.**
2. (Conditional) To create an attribute set, select **New Attribute Set** from the **Attribute Set** drop-down menu.

   An attribute set is a group of attributes that can be exchanged with the trusted provider. For example, you can specify that the local attribute of any attribute in the Liberty profile (such as Informal Name) matches the remote attribute specified at the service provider.

   2a Specify a set name, then click **Next**.

   2b On the Define Attributes page, click **New**.

   2c Select a local attribute.

   2d Specify the name of the remote attribute.

   2e For the namespace, specify **http://schemas.xmlsoap.org/claims**.

   2f Click **OK**.

   2g To add other attributes to the set, repeat Step 2b through Step 2e.

   2h Click **Finish**.
3. Select an attribute set.
4. Select attributes from the **Available** list, and move them to the left side of the page.
5. (Conditional) If you created a new attribute set, it must be enabled for STS.

   For more information, see “Enabling the Attribute Set” on page 490.
6. Click **OK**, then update the Identity Server.

Modifying the User Identification Method

The user identification method specifies how to identify the user.

1. In the Administration Console, click **Devices > Identity Servers > Edit > WS Federation > [Identity Provider] > User Identification.**
2. Select the contract that can be used for authentication. Fill in the following field:

   Satisfies contract: Specifies the contract that is satisfied by the assertion received from the identity provider. WS Federation expects the URI name of the contract to look like a URL, so it rejects all default Access Manager contracts. You must create a contract with a URI that conforms to WS Federation requirements.

   For more information about how to create this contract, see “Creating a New Authentication Contract” on page 488.
3. Specify whether the user can associate (federate) an account at the identity provider (the ADFS server) with an account at Identity Server. Fill in the following field:

   Allow federation: Indicates whether account federation is allowed. Enabling this option assumes that a user account exists at the provider or that a method is provided to create an account that can be associated with the user on subsequent logins. If you do not use this feature, authentication is permitted but is not associated with a particular user account.

4 Select one of the following methods for user identification:

- **Do nothing:** Allows the user to authenticate without creating an association with a user account. This option cannot be used when federation is enabled.
- **Authenticate:** Allows the user to authenticate using a local account.
  - **Allow 'Provisioning':** Provides a button that the user can click to create an account when the authentication credentials do not match an existing account.
- **Provision account:** Allows a new account to be created for the user when the authenticating credentials do not match an existing user. When federation is enabled, the new account is associated with the user and used with subsequent logins. When federation is not enabled, a new account is created every time the user logs in.

  This option requires that you specify a user provisioning method.
- **Attribute matching:** Enables account matching. The service provider can uniquely identify a user in its directory by obtaining specific user attributes sent by the trusted identity provider. This option requires that you specify a user matching method.
  - **Prompt for password on successful match:** Specifies whether to prompt the user for a password when the user's name is matched to an account, to ensure that the account matches.

5 (Conditional) If you selected a method that requires provisioning (**Allow 'Provisioning'** or **Provision account**), click the **Provision settings** icon and create a provisioning method.

For configuration information, see "Defining the User Provisioning Method" on page 420.

6 (Conditional) If you selected **Attribute matching** as the identification method, click the **Attribute Matching settings** icon and create a matching method.

For configuration information, see "Configuring the Attribute Matching Method for Liberty or SAML 2.0" on page 418.

7 Click **OK** twice, then update the Identity Server.

Viewing the WS Identity Provider Metadata

You can view the metadata of the ADFS server, edit it, and view information about the signing certificate.

1 In the Administration Console, click **Devices > Identity Servers > Edit > WS Federation > [Identity Provider] > Metadata.**

The following values need to be configured accurately:

ID: This is provider ID. The ADFS server provides this value to the service provider in the realm parameter in the assertion. You set this value in the **Properties** of the **Trust Policy** on the ADFS server. The label is **Federation Service URI**. The default value is `urn:federation:adatum`.

sloUrl: This is the sign-on URL. This URL is listed in the **Properties** of the **Trust Policy** on the ADFS server. The label is **Federation Services endpoint URL**.

ssoUrl: This is the logout URL. The default value is `https://adfsresource.treyresearch.net/adfs/ls/`. The ADFS server makes no distinction between the login URL and the logout URL.

If the values do not match the ADFS values, you need to edit the metadata.

2 To edit the metadata, click **Edit**. For configuration information, see "Editing the WS Identity Provider Metadata" on page 506.

3 To view information about the signing certificate, click **Certificates**.

4 Click **OK** twice.

Editing the WS Identity Provider Metadata

You can view and edit the metadata of the ADFS server.

1. In the Administration Console, click **Devices > Identity Servers > Edit > WS Federation > [Identity Provider] > Metadata > Edit.**
2. Configure the following fields:

   Provider ID: This is the provider ID. The ADFS server provides this value to the service provider in the realm parameter in the assertion. You set this value in the **Properties** of the **Trust Policy** on the ADFS server. The label is **Federation Service URI**. The default value is `urn:federation:adatum`.

   Sign-on URL: This is the sloUrl. This URL is listed in the **Properties** of the **Trust Policy** on the ADFS server. The label is **Federation Services endpoint URL**.

   Logout URL: This is the ssoUrl. The default value is `https://adfsresource.treyresearch.net/adfs/ls/`. The ADFS server makes no distinction between the login URL and the logout URL.
3. If you need to import a new signing certificate, click the **Browse** button and follow the prompts.
4. To view information about the signing certificate, click **Certificates**.
5. Click **OK** twice, then update the Identity Server.

Modifying the Authentication Card

When you create an identity provider, you must also configure an authentication card. After it is created, you can modify it.

1. In the Administration Console, click **Devices > Identity Servers > Edit > WS Federation > [Identity Provider] > Authentication Card.**
2. Modify the values in one or more of the following fields:

   ID: If you have need to reference this card outside of the Administration Console, specify an alphanumeric value here. If you do not assign a value, the Identity Server creates one for its internal use. The internal value is not persistent. Whenever the Identity Server is rebooted, the value can change. A specified value is persistent.

   Text: Specify the text that is displayed on the card. This value, in combination with the image, indicates to the users the provider they are logging into.

   Image: Specify the image to be displayed on the card. Select the image from the drop-down list. To add an image to the list, click **<Select local image>**.

   Show Card: Determine whether the card is shown to the user, which allows the user to select and use the card for authentication. If this option is not selected, the card is only used when a service provider makes a request for the card.

   Passive Authentication Only: Select this option if you do not want the Identity Server to prompt the user for credentials. If the user has already authenticated and the credentials satisfy the requirements of this contract, the user is passively authenticated. If the user's credentials do not satisfy the requirements of this contract, the user is denied access.
3. Click **OK** twice, then update the Identity Server.

Assertion Validity Window

You can configure the assertion validity time for WS Federation Provider (SP) to accommodate clock skew between a service provider and a SAML identity provider.

To set the assertion validity for WSFed configuration, go to **Devices > Identity Servers > Edit** > **Options**, click new and configure the following property:

Property Type: WSFED ASSERTION VALIDITY

Property Value: Specify the assertion validity time in second

Restart Tomcat by using the following command:

```
/etc/init.d/novell-idp restart
```

Modifying a WS Federation Service Provider

This section explains how to modify a WS Federation service provider after it has been created. "Creating a Service Provider for WS Federation" on page 503 explains the steps required to create the service provider. You can modify the following configuration details:

- "Renaming the Service Provider" on page 507
- "Configuring the Attributes Sent with Authentication" on page 507
- "Modifying the Authentication Response" on page 508
- "Viewing the WS Service Provider Metadata" on page 509
- "Editing the WS Service Provider Metadata" on page 509

Renaming the Service Provider

1. In the Administration Console, click **Devices > Identity Servers > Edit > WS Federation > [Service Provider].**
2. In the **Name** field, specify a new name for the service provider.
3. Click **OK** twice, then update the Identity Server.

Configuring the Attributes Sent with Authentication

When the Identity Server creates its response for the service provider, it uses the attributes listed on the Attributes page. The response needs to contain the attributes that the service provider requires. If you do not own the service provider, you need to contact the administrator of the service provider and negotiate which attributes you need to send in the response. The service provider can then use these attributes to identify the user, to create policies, to match user accounts, or if it allows provisioning, to create a user account on the service provider.

1. In the Administration Console, click **Devices > Identity Servers > Edit > WS Federation > [Service Provider] > Attributes.**
2. (Conditional) To create an attribute set, select **New Attribute Set** from the **Attribute Set** drop-down menu.

   An attribute set is a group of attributes that can be exchanged with the trusted provider. For example, you can specify that the local attribute of any attribute in the Liberty profile (such as Informal Name) matches the remote attribute specified at the service provider.

   2a Specify a set name, then click **Next**.

   2b On the Define Attributes page, click **New**.

2c Select a local attribute.

2d Specify the name of the remote attribute.

2e For the namespace, specify `http://schemas.xmlsoap.org/claims`.

2f Click **OK**.

2g To add other attributes to the set, repeat Step 2b through Step 2e.

2h Click **Finish**.

3 Select an attribute set.

4 Select attributes that you want to send from the **Available** list, and move them to the left side of the page.

5 (Conditional) If you created a new attribute set, it must be enabled for STS.

For more information, see "Enabling the Attribute Set" on page 490.

6 Click **OK**, then update the Identity Server.

Modifying the Authentication Response

When the Identity Server sends its response to the service provider, the response can contain an identifier for the user. If you do not own the service provider, you need to contact the administrator of the service provider and negotiate whether the user needs to be identified and how to do the identification. If the service provider is going to use an attribute for user identification, that attribute needs to be in the attributes sent with authentication. See "Configuring the Attributes Sent with Authentication" on page 507.

To select the user identification method to send in the response:

1 In the Administration Console, click **Devices > Identity Servers > Edit > WS Federation > [Service Provider] > Authentication Response.**

2 For the format, select one of the following:

Unspecified: Specifies that the SAML assertion contains an unspecified name identifier.

E-mail: Specifies that the SAML assertion contains the user's e-mail address for the name identifier.

X509: Specifies that the SAML assertion contains an X.509 certificate for the name identifier.

3 For the value, select an attribute that matches the format. For the Unspecified format, select the attribute that the service provider expects.

The only values available are from the attribute set that you have created for WS Federation.

4 To specify that this Identity Server must authenticate the user, disable the **Use proxied requests** option. When the option is disabled and the Identity Server cannot authenticate the user, the user is denied access.

When this option is enabled, the Identity Server checks to see if other identity providers can satisfy the request. If one or more can, the user is allowed to select which identity provider performs the authentication. If a proxied identity provider performs the authentication, it sends the response to the Identity Server. The Identity Server then sends the response to the service provider.

5 Click **OK** twice, then update the Identity Server.

Viewing the WS Service Provider Metadata

You can view the metadata of the ADFS server, edit it, and view information about the signing certificate.

1 In the Administration Console, click **Devices > Identity Servers > Edit > WS Federation > [Service Provider] > Metadata.**

The following values need to be configured accurately:

ID: This is provider ID. This is the value that the ADFS server provides to the Identity Server in the realm parameter of the query string. This value is specified in the **Properties** of the **Trust Policy** page on the ADFS server. The parameter label is **Federation Service URI**. The default value is `urn:federation:treyresearch`.

sloUrl: This is the sign-on URL. This URL is listed in the **Properties** of the **Trust Policy** on the ADFS server. The label is **Federation Services endpoint URL**. The default value is `https://adfsresource.treyresearch.net/adfs/ls/`.

ssoUrl: This is the logout URL. The default value is `https://adfsresource.treyresearch.net/adfs/ls/`. The ADFS server makes no distinction between the login URL and the logout URL.

If the values do not match the ADFS values, you need to edit the metadata.

2 To edit the metadata, click **Edit**. For configuration information, see "Editing the WS Service Provider Metadata" on page 509.

3 To view information about the signing certificate, click **Certificates**.

4 Click **OK** twice.

Editing the WS Service Provider Metadata

You can view the metadata of the ADFS server and edit metadata.

1 In the Administration Console, click **Devices > Identity Servers > Edit > WS Federation > [Service Provider] > Metadata > Edit.**

2 Configure the following fields:

Provider ID: This is provider ID. This is the value that the ADFS server provides to the Identity Server in the realm parameter of the query string. This value is specified in the **Properties** of the **Trust Policy** page on the ADFS server. The parameter label is **Federation Service URI**. The default value is `urn:federation:treyresearch`.

Sign-on URL: This is the sloUrl. This URL is listed in the **Properties** of the **Trust Policy** on the ADFS server. The label is **Federation Services endpoint URL**. The default value is `https://adfsresource.treyresearch.net/adfs/ls/`.

Logout URL: This is the ssoUrl. The default value is `https://adfsresource.treyresearch.net/adfs/ls/`. The ADFS server makes no distinction between the login URL and the logout URL.

3 If you need to import a new signing certificate, click **Browse** and follow the prompts.

4 To view information about the signing certificate, click **Certificates**.

5 Click **OK** twice, then update the Identity Server.

Configuring STS Attribute Sets

Use the Attribute Set page to select the attribute set or sets that contain attributes the STS can provide to a relying party. An attribute set must be created before you can select it.

When creating an attribute set for the STS, you need to know which protocol you are going to use for the attribute set and select the attributes and namespace appropriate for the protocol.

1. In the Administrations Console, click **Devices** > **Identity Servers** > **Edit** > **WS Federation** > **STS Attribute Sets**.
2. To select a set, move the set from the **Available attribute sets** list to the **Attribute sets** list.

   WS Federation: There is no default attribute set for WS Federation. For information about how to create the set, see "Configuring the Attributes Obtained at Authentication" on page 504 and "Configuring the Attributes Sent with Authentication" on page 507.
3. Click **OK**, then update the Identity Server if you have changed the configuration.

Configuring STS Authentication Methods

Use the Authentication Methods page to select the methods that can be used for authentication at the STS. The methods determine the credentials the user must supply for authentication and the user store that is used to verify the credentials. The WS Federation protocol does not use methods for authentication.

1. In the Administrations Console, click **Devices** > **Identity Servers** > **Edit** > **WS Federation** > **STS Authentication Methods**.
2. To enable a method, move the method from the **Available methods** list to the **Methods** list.

   All the methods that you have defined for the Identity Server appear in the **Available methods** list, but the only default method that works is the Secure Name/Password-Form method. It has been extended so that it knows how to extract name and password information from a managed card that is not backed by a personal card. You can use the Secure Name/Password-Form class to create additional methods for specific user stores.

   You can also create a custom method, if required. For information, see NetIQ Access Manager Developer Tools and Examples (https://www.netiq.com/documentation/access-manager-developer-documentation/).
3. Click **OK**, then update the Identity Server if you have changed the configuration.

Configuring STS Authentication Request

Use the Authentication Request page to select the format for the name identifier that is returned in the SAML assertion. The selected attribute sets determine the values that are available for the formats. If you select a format but do not specify a value, a unique value is generated.

1. In the Administration Console, click **Devices** > **Identity Servers** > **Edit** > **WS Federation** > **STS Authentication Request**.
2. Select one of the following:

   None: Indicates that the SAML assertion does not contain a name identifier.

   Unspecified: Specifies that the SAML assertion contains an unspecified name identifier. For the value, select the attribute that the relying party and the identity provider have agreed to use.

   E-mail: Specifies that the SAML assertion contains the user's e-mail address for the name identifier. For the value, select an e-mail attribute.

X509: Specifies that the SAML assertion contains an X.509 certificate for the name identifier. For the value, select an X.509 attribute.

3 Click **OK**, then update the Identity Server if you have changed the configuration.

Configuring WS-Trust Security Token Service

This section describes WS-Trust Security Token Service (WS-Trust STS) and how to configure it. Topics include:

- "Overview" on page 511
- "Benefits" on page 513
- "Scenarios" on page 513
- "Configuring WS-Trust STS" on page 519
- "Configuring Service Providers" on page 521
- "Configuring Web Service Clients" on page 526
- "Renew Token - Sample Request and Response" on page 527

Overview

Web services are software applications that interact over network by using the Hyper Text Transfer Protocol (HTTP). World Wide Web Consortium (W3C) describes Web services as a standard means of interoperating between software applications running on a variety of platforms and frameworks. A Web service provides a fine-grained service to its client.

Web services use network for communication and data exchange spanning from protected network (intranet) to unprotected network (internet). This demands for security requirements such as client authentication, access control, data integrity, and confidentiality.

You can secure Web services and protect your information against authentication attacks and unauthorized access by using security tokens. A security token contains a set of claims made by a client that includes details such as a user name, password, identity, key, and certificate.

NetIQ Access Manager addresses the need for a secure token mechanism through WS-Trust STS that is based on the WS-Trust protocol. WS-Trust is built on top of the WS-Security specification. It enables Web applications to construct trusted SOAP message exchanges.

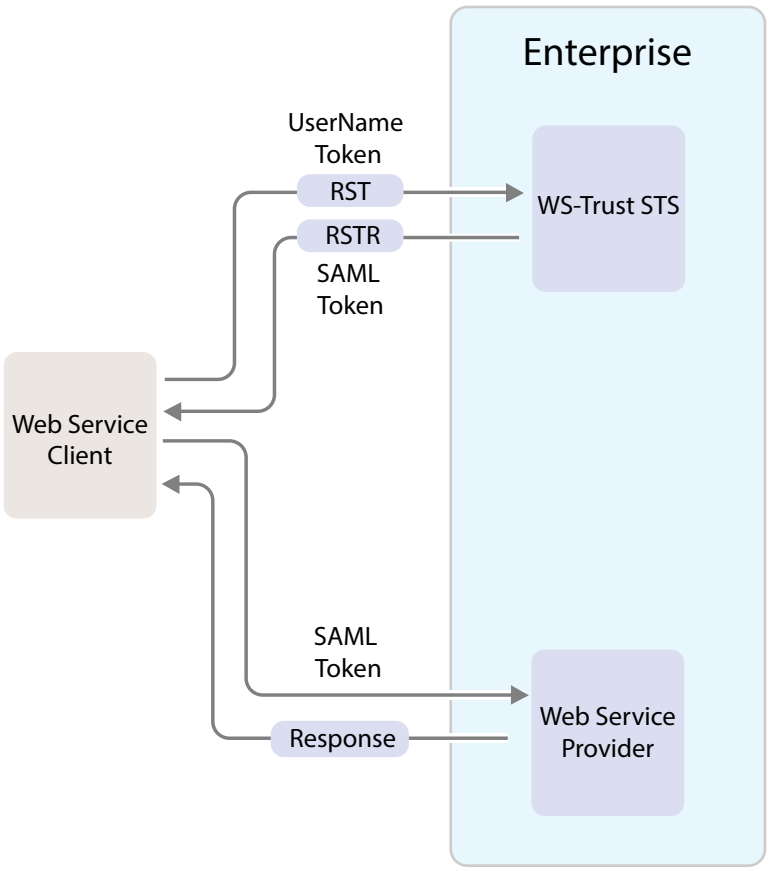
WS-Trust STS provides secure communication and integration between services. It issues and validates security tokens to establish trust relationships between a Web service client and a Web service provider. If the requestor (Web service client) does not have the necessary token to provide required claims to a service, it contacts WS-Trust STS and requests the needed tokens with proper claims. WS-Trust STS may in turn require its own set of claims for authenticating and authorizing the request for security tokens.

How WS-Trust STS Works

WS-Trust STS allows secure identity propagation and token exchange between Web services. It provides a standard framework for requesting and returning security tokens by using Request Security Token (RST) and Request Security Token Response (RSTR) messages. RST provides the means for requesting a security token from WS-Trust STS. RSTR contains the requested token, claims, and other related information.

The Web service client provides its credentials to WS-Trust STS and gets a SAML token from WS-Trust STS. A trust is established between the Web service provider and WS-Trust STS. The Web service client presents the token from WS-Trust STS to the Web service provider. The Web service provider validates if the token has been issued from WS-Trust STS and grants access to the required service.

The following diagram illustrates an example of how WS-Trust STS facilitates a secure communication between a Web service client and a Web service provider through security tokens:



WS-Trust STS is designed to interoperate with many different Web Service environments and supports SOAP and WS-Trust specifications.

Web services must implement the protocol defined in the WS-Trust 1.3 or 1.4 specification by making assertions based on evidence that it trusts, to whoever trusts it, or to specific recipients. For more information about WS-Trust specification, see WS-Trust 1.3 Specification and WS-Trust 1.4 Specification.

The following table lists supported SOAP and WS-Trust versions and corresponding namespaces:

Specification	Version	Namespace
Soap	1.1	http://schemas.xmlsoap.org/soap/envelope/
	1.2	http://www.w3.org/2003/05/soap-envelope
WS-Trust	1.3	http://docs.oasis-open.org/ws-sx/ws-trust/200512/
	1.4	http://docs.oasis-open.org/ws-sx/ws-trust/200802

NOTE

- WS-Trust STS supports SAML tokens in addition to usernametokens.

  WS-Trust STS can issue both SAML 1.1 and SAML 2.0 based tokens.
- WS-Trust STS supports issuing, validating and renewing tokens. This release does not support canceling tokens.
- Web service clients and Web service providers must be in the same domain. This release does not support multiple domains.

Benefits

WS-Trust STS offers the following benefits:

- Enables the secure exchange of SOAP messages among Web services.
- Facilitates identity delegation (through ActAs) and impersonation (through OnBehalfOf) where an authenticated user is granted access to downstream Web services.
- Reduces complexity for the Web service consumer as the Web service consumer does not need token specific knowledge.

Scenarios

This section describes the basic scenarios supported by WS-Trust STS.

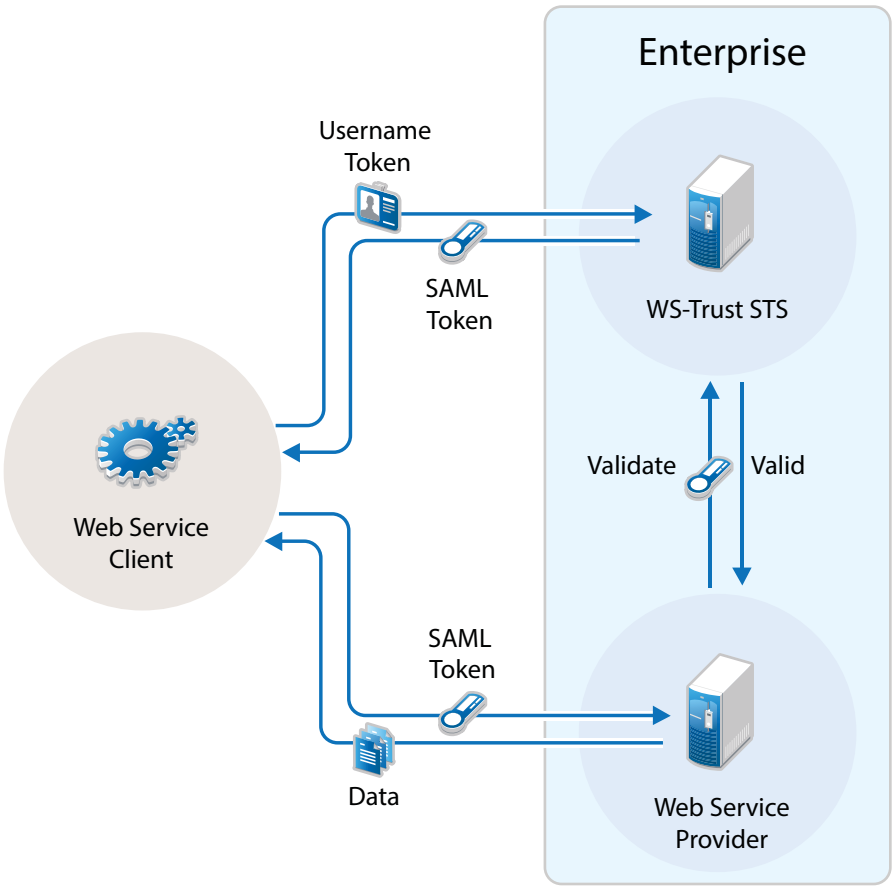
- "Scenario 1: Web Service Client Communicating with Token Protected Web Service Provider" on page 513
- "Scenario 2: Web Single Sign-On and STS" on page 514
- "Scenario 3: Identity Delegation and Impersonation" on page 515
- "Renewing a Token" on page 517
- "Authentication by Using SAML Tokens" on page 518

Scenario 1: Web Service Client Communicating with Token Protected Web Service Provider

In this scenario, a Web service client situated outside the enterprise tries to access a Web service provider hosted inside the enterprise.

This process consists of requesting a token by means of the request-response message pairs of a Request Security Token (RST) and a Request Security Token Response (RSTR). The tokens are included in SOAP messages.

The following diagram illustrates this scenario:

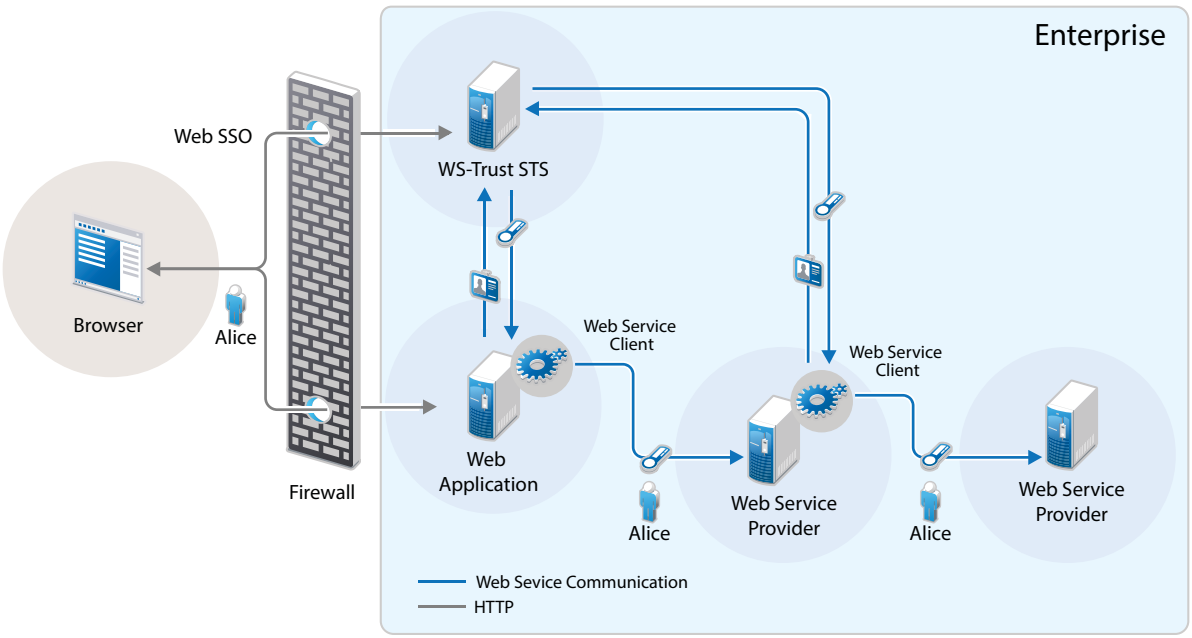


1. A Web service client, which is outside the enterprise, sends its credentials to WS-Trust STS and request for the security token through RST.
2. WS-Trust STS verifies the client's credentials and then issues a security token (SAML token) through RSTR.

   The Web service client caches the security token and then uses it in multiple requests to the Web service provider.
3. The Web service client presents the token to the Web service provider.
4. The Web service provider validates the token and sends the response to the Web service client.

Scenario 2: Web Single Sign-On and STS

In this scenario, a Web service client that resides as part of a Web application within an enterprise tries to access services from other Web service providers of the same enterprise. A user named Alice accesses to the Web application through a browser and needs single sign-on access to other applications.



1. A Web application sends a single sign-on authentication request to WS-Trust STS on behalf of Alice.

   The Web application is residing within the enterprise.
2. The Web application passes the credentials corresponding to the single sign-on session to the Web service client.
3. The Web service client requests for security token by using the passed on credentials.
4. WS-Trust STS verifies the credentials. After checking the credentials, it verifies if the Web service provider for which the token has been requested for is a trusted service provider. Then it issues a security token consumable by the service provider.
5. The Web service client residing within the Web application presents the token to the Web service provider. The Web service client caches the security token and then uses it in multiple requests to the Web service provider.
6. The Web service provider validates the token and sends the response to the Web service client residing within the Web application.

The tokens are included in SOAP messages.

Scenario 3: Identity Delegation and Impersonation

In this scenario, a Web service provider requests services from other Web service providers.

The following use-case explains this scenario:

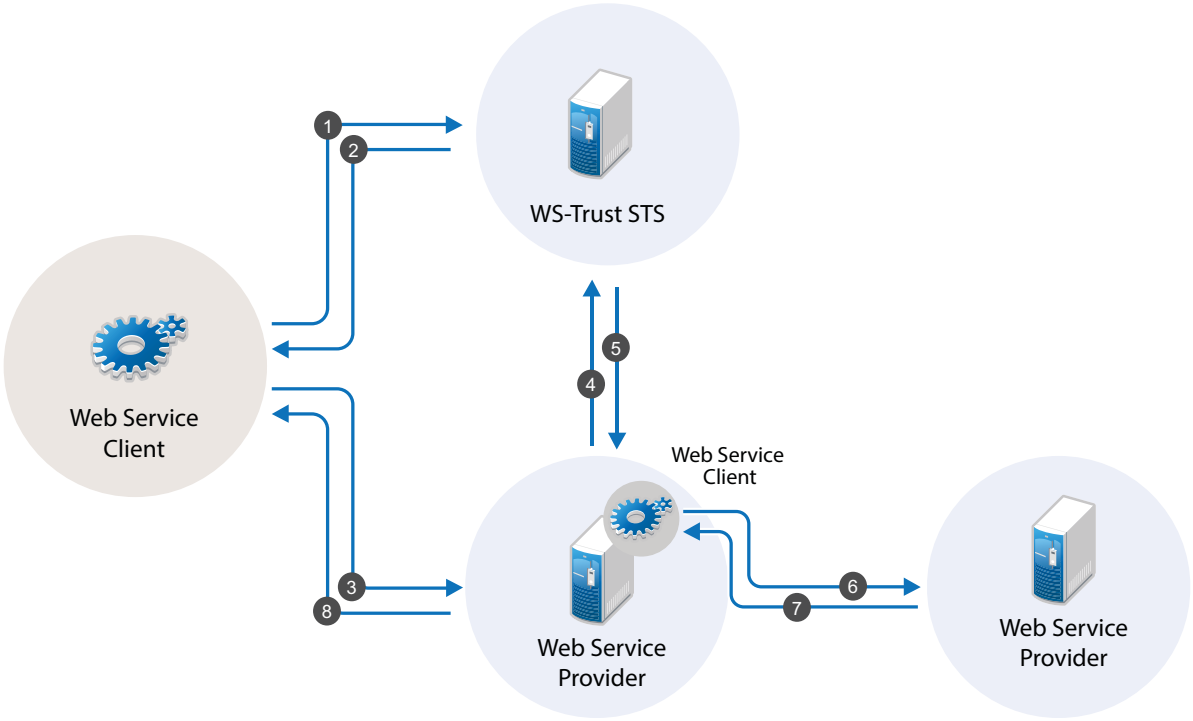
There are two Web service providers called Service1 and Service2 providing some fine-grained service. Both Service1 and Service2 require authentication and trust WS-Trust STS. A Web service client tries to access Service1 and requests for the service. To fulfill the request, Service1 needs to access the service of Service2. Service1 can request a token from WS-Trust STS to access Service2. This exchange of information happens through security tokens embedded in SOAP messages.

This chaining of service request can be any number of nodes based on the implementation.

Requests for tokens can be made in two ways:

- **By using the ActAs element (Identity Delegation):** The ActAs element is used for delegated requests. Delegated scenarios require composite delegation, where the final recipient of the issued token can see the entire delegation chain. ActAs is commonly used in multi-tiered systems to authenticate and pass information about identities between the tiers without having to pass this information at the application or business logic layer.
- **By using the OnBehalfOf element (Impersonation):** The OnBehalfOf element is used for proxy requests. OnBehalfOf is used when the identity of only the original client is important. The final recipient of the issued token can only see claims about the original client. The information about intermediaries is not preserved.

The following diagram illustrates the workflow:



The following workflow explains the ActAs scenario:

1. The Web service client sends a RST to WS-Trust STS for its authentication.
2. WS-Trust STS returns a SAML token to the client in the RSTR. Let us refer to this SAML token as token1. The subject of this SAML token is `client`.
3. The client forwards the token1 with its SOAP request to Service1.
4. Then Service1 sends a RST to WS-Trust STS again authenticating itself to the STS. This time the RST contains the token1 inside the ActAs element.
5. WS-Trust STS issues a SAML token (referred to as token2). The subject of this token is `Service1`. It contains an attribute called `ActAs` with the value of `client`.
6. Service1 sends token2 to Service2. By processing token2, Service2 understands that the original requester is `client` and Service1 is acting as the original requester.
7. Service2 sends the response to Service1.
8. Service1 forwards the response to the client.

The following workflow explains the OnBehalfOf scenario:

1. The Web service client sends a RST to WS-Trust STS for its authentication.
2. WS-Trust STS returns a SAML token to the client in the RSTR. Let us refer to this SAML token as token1. The subject of this SAML token is `client`.
3. The client forwards the token1 with its SOAP request to Service1.
4. Then Service1 sends a RST to WS-Trust STS again authenticating itself to the STS. This time the RST contains the token1 inside the OnBehalfOf element.
5. WS-Trust STS issues a SAML token (referred to as token2). The subject of this token is `client`.
6. Service1 sends token2 to Service2. Service2 understands that the original requester is `client` but cannot see the details of Service1.
7. Service2 sends the response to Service1.
8. Service1 forwards the response to the client.

IMPORTANT: Starting from Access Manager 4.0 SP1 release, the default binding supported is SOAP 1.2. If you want to use SOAP 1.1 instead, perform the following steps on all instances of the Identity Server:

1. Traverse to the `/opt/novell/nam/idp/webapps/nidp/WEB-INF` folder and edit the `sun-jaxws.xml` file.
2. Remove all instances of bindings from the endpoints in the `sun-jaxws.xml` file and save the changes. A binding is represented by the following line in this file:

   ```
   binding="http://java.sun.com/xml/ns/jaxws/2003/05/soap/bindings/HTTP/"
   ```

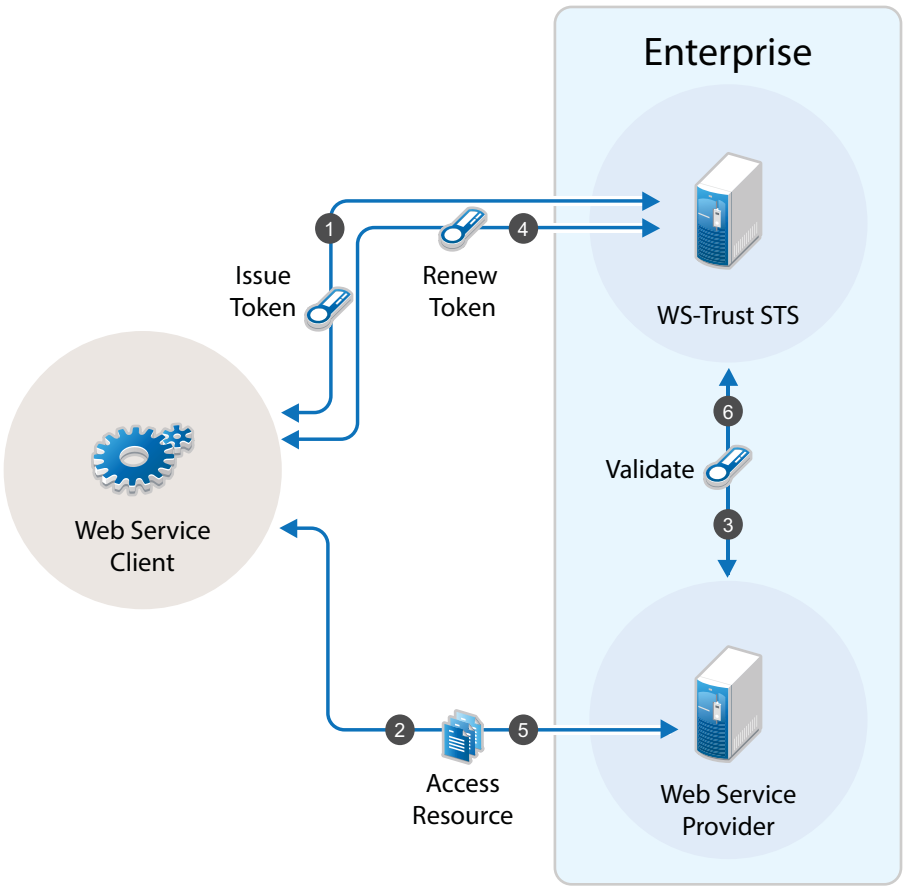
3. Restart the Identity Server using the `/etc/init.d/novell-idp restart` command.

Renewing a Token

The renew token operation helps in renewing a token issued by WS-Trust Security Token Service(STS). Only a token that is issued by an Identity Server that is part of the same cluster can be renewed. Tokens issued by a different Identity Server in a different cluster or by a third-party STS cannot be renewed.

Each token generated by the STS is valid for the duration specified using the **Token Lifetime** setting. A token can be renewed only before lapse of the expiry period. For example: if the Token Lifetime has been specified as 180 seconds, token renew operation will succeed only till the 179th second.

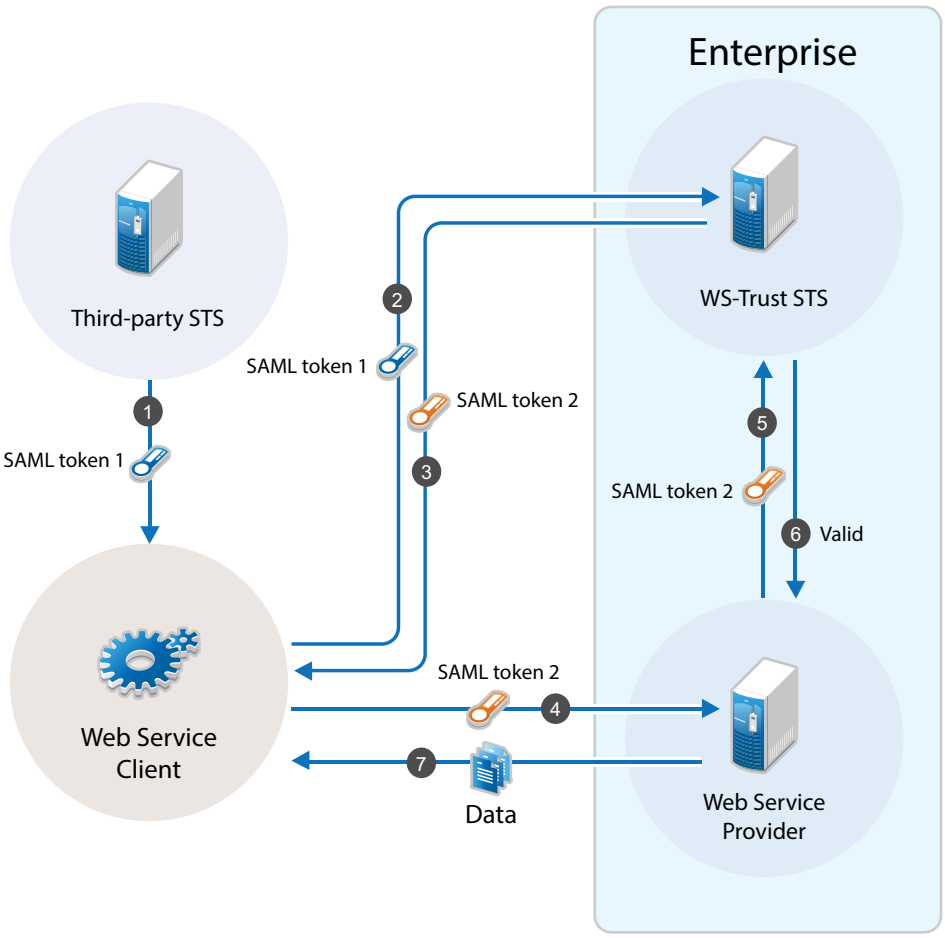
Workflow:



1. The Web service client sends a RST to WS-Trust STS for its authentication and WS-Trust STS returns a SAML token to the client in the RSTR
2. Web Service Provider uses the SAML token from STS and requests access to resources hosted on the Web Service provider.
3. The Web Service Provider validates the SAML token and provides access to the resources.
4. When the token is nearing expiry, the Web service client sends a RST to WS-Trust STS to renew the token previously issued. The STS renews the validity of the token and sends a renewed token to the Web Service client for any further requests.
5. Web Service client uses the renewed SAML token from STS and requests access to resources hosted on the Web Service provider.

Authentication by Using SAML Tokens

WS-Trust STS accepts SAML tokens issued by a third-party STS for authentication. The tokens can be in SAML 1.1 or SAML 2.0.



Workflow:

1. The Web service client sends a RST to third-party STS. The third-party STS returns a SAML token to the client in the RSTR.
2. The Web Service client uses SAML token issued by the third-party STS and requests WS-Trust STS to grant access to resources hosted on the Web Service Provider.
3. WS-Trust STS authenticates the client using the third-party SAML token and issues a new SAML token.
4. The Web Service client uses this new SAML token to get access to resources hosted on the Web Service Provider.
5. The Web Service Provider validates the SAML token with WS-Trust STS.
6. The Web Service client can access the resources on the Web Service Provider if the SAML token is valid.

Configuring WS-Trust STS

Before a Web service can invoke operations on STS, you must enable WS-Trust and configure it in Access Manager. This section discusses the following topics:

- “Enabling WS-Trust” on page 520
- “Configuring Access Manager for WS-Trust STS” on page 520
- “Viewing STS Service Details” on page 520

Enabling WS-Trust

Access Manager ships with only SAML 1.1, Liberty, and SAML 2.0 enabled by default. To use the WS-Trust protocol, you must enable it on the Identity Server.

To enable WS-Trust, perform the following steps:

1. In the Administration Console, click **Devices** > **Identity Servers** > **Edit**.
2. In the **Enabled Protocols** section, select **WS-Trust**.
3. Click **OK**.
4. Update the Identity Server.

Configuring Access Manager for WS-Trust STS

To configure WS-Trust STS, perform the following steps:

1. In the Administration Console, click **Devices** > **Identity Servers** > **Edit**.
2. Select **WS-Trust** > **STS Configuration**.
3. Specify the following details:

   Token Lifetime: Specify the duration in seconds for which the token is valid. The default value is 360 seconds.

   Token Issuer: Specify the name of the issuer of the authentication token.

   Authentication Methods: Select methods that can be used for authentication at STS for WS-Trust.

   Select and move methods from **Available Authentication Methods** to **Selected Authentication Methods**.

   Tokens: Select the type of tokens that can be issued for authentication at STS for WS-Trust. SAML 1.1 and SAML 2.0 tokens are supported. If you select both token types, the token type configured in the service provider is returned.
4. Click **Apply**.

Viewing STS Service Details

EndPoint URL is the SOAP endpoint of STS. The Web service client and Web service provider must be configured to these endpoints.

In the Administration Console under **Devices** > **Identity Servers** > **Edit** > **WS-Trust** > **EndPoint Summary**, you can view the following STS EndPoint details:

Service Name: The name of the STS service endpoint that is configured in the Web service client.

Port Name: The port that STS implements. This is configured in the Web service client.

MEX EndPoint URI: The MetadataExchange endpoint of STS.

WSDL of STS Username authentication: WSDL location for username authentication. This file is used by applications that use the token service with username authentication.

WSDL of STS SAML authentication: WSDL location for SAML. This file is used by applications that use the token service with SAML authentication.

Configuring Service Providers

You require to configure Web service providers to accept tokens issued by an STS. The Web service provider uses an IssuedToken policy for the same. The IssuedToken policy is wrapped in WSDL. For a sample policy, see “A Sample WS-Policy for Web Service Providers” on page 525.

Configuring a service provider includes adding a service provider domain and then adding a service provider in a configured domain. Access Manager also allows you to modify and delete configured service provider domains and service providers.

- “Adding a Domain and Assigning WS-Trust Operations” on page 521
- “Adding Web Service Providers” on page 521
- “Managing Service Provider Domains” on page 523
- “Managing Service Providers” on page 524
- “Modifying Service Providers” on page 524
- “A Sample WS-Policy for Web Service Providers” on page 525

Adding a Domain and Assigning WS-Trust Operations

1. In the Administration Console, click **Devices** > **Identity Servers** > **Edit** > **WS-Trust** > **Service Provider Domain**.
2. Click **New > General** to create a general domain. Selecting **New > Office 365** creates an Office 365 domain that can be configured for active authentication. For details on creating an Office 365 domain, see “Configuring an Office 365 Domain By Using WS-Trust Protocol” on page 576
3. Specify the following details:

   Name: Specify a name for the domain.

   WS-Trust Operations: Select operations in **Available operations** that WS-Trust STS performs for tokens and move these to **Selected operations**.

   The available operations are Issue, Validate, OnBehalfOf, ActAs and Renew.

   If you select OnBehalfOf and Act As the Available operations, additional configuration is required. For more information, see “Adding Policy for ActAs and OnBehalfOf” on page 522
4. Click **Finish**. Continue with creation of a trusted Service Provider. For more information, see Adding Web Service Providers

Adding Web Service Providers

This section discusses how to add service providers for WS-Trust STS. Adding a service provider includes adding service provider EndPoint URL, configuring trust certificates, selecting token types, and customizing attributes.

Perform the following steps:

1. In the Administration Console, click **Devices** > **Identity Servers** > **Edit** > **WS-Trust** > **Service Provider Domain**.
2. Select the domain under which you want to configure a service provider.
3. Click **Service Provider** > **New**.
4. Specify the following details:

   Name: Specify a name for the service provider.

   Endpoint: Specify the SOAP endpoint location at the service provider to which SOAP messages are sent.

Token Type: Select the type of token that the service provider will accept or validate.

Encrypt Proof Token Using: Import a certificate from the file system or paste content of the certificate here. This certificate must be configured in the Web service provider and is used for creating the subject confirmation in the SAML token.

5 Click **Finish**.

6 Select the Service Provider to define the Attributes and Authentication Response. For more information, see "Modifying Service Providers" on page 524

Enabling Delegation and Impersonation

By default, ActAs and OnBehalfOf requests are disabled in the Access Manager Identity Server. To enable delegation and impersonation, you must enable ActAs and OnBehalfOf by performing the following steps:

1 Go to **WS-Trust** > **Service Provider Domain**.

2 Click the service provider domain name for which you want to enable ActAs and OnBehalfOf operations.

3 Under **WS Trust Operations**, select **ActAs** and **OnBehalfOf** in **Available operations** and move to **Selected operations**.

4 Click **OK**.

These operations are restricted to a set of privileged user accounts defined in the policy. You need to configure the allowed user accounts, who can perform ActAs and OnBehalfOf operations, in the `nidconfig.properties` file of each Identity Server installation. For more information, see "Adding Policy for ActAs and OnBehalfOf" on page 522

Configuring ActAs to Lookup Multiple User Stores

For ActAs, the username on behalf of whom a client requests for a token must be present in the user store (eDirectory). The default implementation checks for this user only in the default user store. If you want to search the user in a different user store, perform the following steps:

1 In the Administration Console, click **Devices** > **Identity Server** > **Edit** > **Local** > **Classes**.

2 Click **New** and specify the following details:

Display name: Specify Find_By_Username

Java class: Select Other

Java class path: Specify com.novell.nidp.authentication.local.UserNameAuthenticationClass

3 Click **Next** > **Finish**.

4 Go to **Local** > **Methods**.

5 Click **New** and select the Find_By_Username class.

For more information about how to configure an authentication method, see "Configuring Authentication Methods" on page 287.

6 Go to **WS-Trust** > **STS Configuration**. Move this authentication method in the **Selected Authentication Methods** from **Available Authentication Methods**.

Adding Policy for ActAs and OnBehalfOf

You must add an policy to allow ActAs and OnBehalfOf operations. The default policy looks for a configuration of allowed user names from the `nidpconfig.properties` file. Allowed usernames are the user accounts that the intermediate Web service provider uses to authenticate with STS when

sending a request with ActAs or OnBehalfOf elements.For ActAs and OnBehalfOf, you must specify multiple username values separated with comma. If no value is specified, ActAs and OnBehalfOf are denied.

1 In the Administration Console, click **Devices > Identity Servers > Edit** > **Options**.

2 Click **New**.

3 Set the following properties based on your requirement:

Property Type	Property Value
WSTRUST AUTHORIZATION ALLOWED ACTAS VALUES	Specify the user names who can perform ACTas operations. Allowed user names are the user accounts that the intermediate web service provider uses to authenticate with STS when sending a request with Actas elements.
WSTRUST AUTHORIZATION ALLOWED ONBEHALF VALUES	Specify the user names who can perform OnBehalfOf operations. Allowed user names are the user accounts that the intermediate web service provider uses to authenticate with STS when sending a request with OnBehalfOf elements.
WSTRUST AUTHORIZATION ALLOWED VALUES	Specify the user names who can perform both Actas and onBehalfOf operations.

4 Click **OK** > **Apply**.

5 Restart the Identity Server by running the following command:

After upgrading Access manager, the configuration is set to default values. You must reconfigure the details after each upgrade.

```
/etc/init.d/novell-idp restart
```

Managing Service Provider Domains

The WS-Trust page allows you to create, modify, and delete service provider domains. This page lists all configured service provider domains.

1 In the Administration Console, click **Devices** > **Identity Servers** > **Edit** > **WS-Trust** > **Service Provider Domains**.

The list of all configured service provider domains is displayed.

2 Select one of the following actions:

- **New:** Select **New > General** to create a general domain. Selecting **New > Office 365** creates a domain that can be configured for single sign-on to Office 365 services. For more on creating Office 365 domain, see “Adding a Domain and Assigning WS-Trust Operations” on page 521.
- **Delete:** Deletes the selected service provider domain.

3 Click **OK**, then update the Identity Server.

4 Select the Service Provider domain to modify the following details:

- **Name:** Modify the name of the service provider domain.
- **WS Trust Operations:** Modify the list of selected WS-Trust operations.

5 Click **OK**.

Managing Service Providers

Access Manager allows you to you to create, modify, and delete trusted service providers. The Service Providers page lists all configured service provider.

1 In the Administration Console, click **Devices** > **Identity Servers** > **Edit** > **WS-Trust**> **Service Provider Domains**> [name of the service provider domain] > **Service Providers**.

The list of all configured service provider for the selected domain is displayed.

2 Select one of the following actions:

- **New:** Launches the Create a Service Provider page. For more information, see "Adding Web Service Providers" on page 521.
- **Delete:** Deletes the selected service providers.

3 Click **OK**.

Modifying Service Providers

1 In the Administration Console, click **Devices** > **Identity Servers** > **Edit** > **WS-Trust**> **Service Provider Domains**> [name of the service provider domain] > **Service Providers**.

The list of all configured service provider for the selected domain is displayed.

2 Click the name of the service provider you want to edit.

Configuration > Trust

You can modify the following details:

- Name
- Endpoint
- Token Type
- Encrypt Proof Token Using

For more information about these fields, see "Adding Web Service Providers" on page 521.

Configuration > Attributes

- Select the Attribute Set and move attributes from the Available list to the **Send with Authentication** pane. This indicates the attributes that you want sent in an assertion to the service provider.

Configuration > Authentication Response

- Specify a value for the name identifier.
  - The persistent and transient formats are generated automatically. For the others, you can select an attribute. The available attributes depend upon the attributes that you have selected to send with authentication (see Configuring the Attributes Obtained at Authentication). If you do not select a value for the E-mail, Kerberos, X509, or Unspecified format, a unique value is automatically generated.

IMPORTANT: In Access Manager 4.0 SP1, the SAML tokens with Name Identifier value other than username do not support ActAs, OnBehalfOf and SAML authentication operations.

---

3 Click **Apply**.

A Sample WS-Policy for Web Service Providers

You must modify WSDL of a Web service provider to include IssuedTokenPolicy that points to Access Manager WS-Trust STS. To modify WSDL, you require to add a WS-Policy with IssuedTokenElement. The following is a sample configuration:

```
<wsp:Policy wsu:Id="<policy_name>">
        <wsp:ExactlyOne>
            <wsp:All>
                <wsaws:UsingAddressing xmlns:wsaws="http://www.w3.org/2006/05/
addressing/wsdl" wsp:Optional="false"/>
              <sc:KeyStore wspp:visibility="private" alias="xws-security-server"/>
                <sp:SymmetricBinding>
                    <wsp:Policy>
                        <sp:ProtectionToken>
                            <wsp:Policy>
                                <sp:IssuedToken sp:IncludeToken="http://
schemas.xmlsoap.org/ws/2005/07/securitypolicy/IncludeToken/AlwaysToRecipient">
                                    <sp:RequestSecurityTokenTemplate>
                                      <t:TokenType>http://docs.oasis-open.org/wss/
oasis-wss-saml-token-profile-1.1#SAMLV1.1</t:TokenType>
                                        <t:KeyType>http://schemas.xmlsoap.org/ws/
2005/02/trust/SymmetricKey</t:KeyType>
                                        <t:KeySize>256</t:KeySize>
                                    </sp:RequestSecurityTokenTemplate>
                                    <wsp:Policy>
                                        <sp:RequireInternalReference/>
                                    </wsp:Policy>
                                    <sp:Issuer>
                                        <wsaws:Address>https://namtest.com:8443/
nidp/wstrust/sts</wsaws:Address>
                                        <wsaws:Metadata>
                                            <wsx:Metadata>
                                                <wsx:MetadataSection>
                                                    <wsx:MetadataReference>
                                                        <wsaws:Address>https://
namtest.com:8443/nidp/wstrust/sts/mex</wsaws:Address>
                                                    </wsx:MetadataReference>
                                                </wsx:MetadataSection>
                                            </wsx:Metadata>
                                        </wsaws:Metadata>
                                    </sp:Issuer>
                                </sp:IssuedToken>
                            </wsp:Policy>
                        </sp:ProtectionToken>
                        <sp:Layout>
                            <wsp:Policy>
                                <sp:Lax/>
                            </wsp:Policy>
                        </sp:Layout>
                        <sp:IncludeTimestamp/>
                        <sp:OnlySignEntireHeadersAndBody/>
                        <sp:AlgorithmSuite>
                            <wsp:Policy>
                                <sp:Basic128/>
                            </wsp:Policy>
                        </sp:AlgorithmSuite>
                    </wsp:Policy>
                </sp:SymmetricBinding>
                <sp:Wss11>
```

```
                    <wsp:Policy>
                        <sp:MustSupportRefIssuerSerial/>
                        <sp:MustSupportRefThumbprint/>
                        <sp:MustSupportRefEncryptedKey/>
                    </wsp:Policy>
                </sp:Wss11>
                <sp:Trust10>
                    <wsp:Policy>
                        <sp:MustSupportIssuedTokens/>
                        <sp:RequireClientEntropy/>
                        <sp:RequireServerEntropy/>
                    </wsp:Policy>
                </sp:Trust10>
            </wsp:All>
        </wsp:ExactlyOne>
    </wsp:Policy>
```

Configuring Web Service Clients

Access Manager WS-Trust STS can be accessed from various Web service clients. The following sections provide example configurations and sample code snippets for CXF-based and Metro-based Web service clients:

- “Configuring Apache CXF-based Web Service Clients” on page 526
- “Configuring Metro-based Web Service Clients” on page 527

Configuring Apache CXF-based Web Service Clients

You can configure CXF-based Web service clients either programmatically or through XML configuration files. Below is a sample XML configuration. Add the following features to `cxf.xml` under the top-level beans section:

```
<cxf:bus>
    <cxf:features>
      <cxf:logging />
      <wsa:addressing />
    </cxf:features>
  </cxf:bus>
```

Define the STS client with its properties as follows:

```
<jaxws:client name="{<your webservice target namespace>}WebServicePort"
    createdFromAPI="true">
    <jaxws:properties>
<entry key="ws-security.sts.client">
        <bean class="org.apache.cxf.ws.security.trust.STSClient">
          <constructor-arg ref="cxf" />
          <property name="wsdlLocation"
            value="https://<your idp base url>nidp/wstrust/sts?wsdl" />
          <property name="serviceName" value="{http://www.netiq.com/nam-4-0/
wstrust}SecurityTokenService" />
          <property name="endpointName" value="{http://www.netiq.com/nam-4-0/
wstrust}STS_Port" />

          <property name="wspNamespace" value="http://schemas.xmlsoap.org/ws/2004/
09/policy" />
          <property name="properties">
```

```
            <map>
            <entry key="ws-security.username" value="<username to connect to idp>"
/>
              <entry key="ws-security.password" value="<password>" />
              <entry key="ws-security.encryption.properties"
value="clientKeystore.properties" />
              <entry key="ws-security.encryption.username" value="mystskey" />
              <entry key="soap.force.doclit.bare" value="true" />
              <entry key="soap.no.validate.parts" value="true" />
            </map>
          </property>
        </bean>
      </entry>
</jaxws:clien>
```

You can configure `ws-security.callback-handler` to provide username and password programmatically. You can also configure global sts-client in `cxf.xml` that can be used across multiple Web services.

For more information about configuring Apache CXF-based Web service clients, see Apache CFX (http://cxf.apache.org/docs/ws-trust.html).

Configuring Metro-based Web Service Clients

You can configure Metro-based clients through NetBeans (an integrated development environment).

1. Create a Web service client project in NetBeans.
2. Right click the project and click **Create Web Service Client** to create a STS client. Point the WSDL to http://<name of the identity provider server>:<port>/nidp/wstrust/sts?wsdl.
3. Configure the username and password to access WS-Trust STS.

   The user configured needs to get authenticated into Access Manager password-based authentication classes. You can also configure the Callback-based configuration in NetBeans to provide username and passwords dynamically.
4. When you create a Web service client for your Web service, which is configured for STS-issued tokens, you need to specify the endpoint URL of WS-Trust STS in the Web service client properties. You can specify this in NetBeans by right clicking **Web Service References**> **Web Service** and selecting **Secure Token Service**.

For more information about configuring Metro-based Web service clients, see *To Specify an STS on the Service Side* and *To Specify an STS on the Client Side* in Configuring A Secure Token Service (STS).

Renew Token - Sample Request and Response

- “Renew Token - Sample Request” on page 528
- “Renew Token - Sample Response” on page 530

Renew Token - Sample Request

```
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope">
   <soap:Header>
      <Action xmlns="http://www.w3.org/2005/08/addressing">http://docs.oasis-
open.org/ws-sx/ws-trust/200512/RST/Renew</Action>
      <MessageID xmlns="http://www.w3.org/2005/08/addressing">urn:uuid:9cfedcee-
2ebf-47e0-a24a-45281d785136</MessageID>
      <To xmlns="http://www.w3.org/2005/08/addressing">https://
namsb.blr.novell.com:443/nidp/wstrust/sts</To>
      <ReplyTo xmlns="http://www.w3.org/2005/08/addressing">
         <Address>http://www.w3.org/2005/08/addressing/anonymous</Address>
      </ReplyTo>
      <wsse:Security soap:mustUnderstand="1" xmlns:wsse="http://docs.oasis-
open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd" xmlns:wsu="http:/
/docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd">
         <wsu:Timestamp wsu:Id="TS-1">
            <wsu:Created>2014-02-10T23:36:42Z</wsu:Created>
            <wsu:Expires>2014-02-10T24:36:42Z</wsu:Expires>
         </wsu:Timestamp>
         <wsse:UsernameToken wsu:Id="UsernameToken-2">
            <wsse:Username>admin</wsse:Username>
            <wsse:Password Type="http://docs.oasis-open.org/wss/2004/01/oasis-
200401-wss-username-token-profile-1.0#PasswordText">novell</wsse:Password>
         </wsse:UsernameToken>
      </wsse:Security>
   </soap:Header>
   <soap:Body>
      <wst:RequestSecurityToken xmlns:wst="http://docs.oasis-open.org/ws-sx/ws-
trust/200512" xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
wssecurity-secext-1.0.xsd">
         <wst:RequestType>http://docs.oasis-open.org/ws-sx/ws-trust/200512/Renew</
wst:RequestType>
         <wst:TokenType>urn:oasis:names:tc:SAML:2.0:assertion</wst:TokenType>
         <wst:KeyType>http://docs.oasis-open.org/ws-sx/ws-trust/200512/
SymmetricKey</wst:KeyType>
         <wst:Entropy>
            <wst:BinarySecret Type="http://docs.oasis-open.org/ws-sx/ws-trust/
200512/Nonce">2O0dAaqhrBJqbouiQTf7D2pXtXRo36Wi/yswGeoq7iQ=</wst:BinarySecret>
         </wst:Entropy>
         <wst:RenewTarget>
            <saml2:Assertion ID="nstsa657b5f4-9bf0-45b7-9875-07eeb6d65196"
IssueInstant="2014-05-26T10:33:50.564Z" Version="2.0" xmlns:ds="http://www.w3.org/
2000/09/xmldsig#" xmlns:exc14n="http://www.w3.org/2001/10/xml-exc-c14n#"
xmlns:saml2="urn:oasis:names:tc:SAML:2.0:assertion" xmlns:xenc="http://www.w3.org/
2001/04/xmlenc#" xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:ns10="http://
www.w3.org/2000/09/xmldsig#" xmlns:ns13="http://www.w3.org/2001/10/xml-exc-c14n#"
xmlns:ns3="http://docs.oasis-open.org/wss/oasis-wss-wssecurity-secext-1.1.xsd"
xmlns:ns5="http://docs.oasis-open.org/ws-sx/ws-trust/200512/" xmlns:ns9="http://
schemas.xmlsoap.org/ws/2006/02/addressingidentity" xmlns:sc="http://docs.oasis-
open.org/ws-sx/ws-secureconversation/200512" xmlns:trust="http://docs.oasis-
open.org/ws-sx/ws-trust/200512" xmlns:wsa="http://www.w3.org/2005/08/addressing"
xmlns:wsp="http://schemas.xmlsoap.org/ws/2004/09/policy" xmlns:wst="http://
schemas.xmlsoap.org/ws/2005/02/trust" xmlns:wsu="http://docs.oasis-open.org/wss/
2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd" xmlns:S="http://www.w3.org/
2003/05/soap-envelope" xmlns:wsse11="http://docs.oasis-open.org/wss/oasis-wss-
wssecurity-secext-1.1.xsd">
               <saml2:Issuer>https://namsb.blr.novell.com/nidp/wstrust/sts</
saml2:Issuer>
               <ds:Signature>
```

```
                  <ds:SignedInfo>
                     <ds:CanonicalizationMethod Algorithm="http://www.w3.org/2001/
10/xml-exc-c14n#"/>
                     <ds:SignatureMethod Algorithm="http://www.w3.org/2000/09/
xmldsig#rsa-sha1"/>
                    <ds:Reference URI="#nstsa657b5f4-9bf0-45b7-9875-07eeb6d65196">
                        <ds:Transforms>
                           <ds:Transform Algorithm="http://www.w3.org/2000/09/
xmldsig#enveloped-signature"/>
                           <ds:Transform Algorithm="http://www.w3.org/2001/10/xml-
exc-c14n#"/>
                        </ds:Transforms>
                        <ds:DigestMethod Algorithm="http://www.w3.org/2000/09/
xmldsig#sha1"/>
                        <ds:DigestValue>Z3S4qxz2wRv0k5np2R6ENkIF9pk=</
ds:DigestValue>
                     </ds:Reference>
                  </ds:SignedInfo>

<ds:SignatureValue>ZxeqeuD7NXfNRPiaYY3v2Nfo9vTx+ceASiAFBDzOfaWGczHBT0eYU+AQM99vdX1
GCBCdWqO9qQR8
2WP71mzREC6ndg+8g/zJ6UH+Jzsf05hIxCAu7d7fg5qP5/BP++x8vUlpUQ32D8daxx+GIwuZjlOs
8KhdbgLReYSWyX6PV0UbjbnAtDFaBTTJ5lpEqHdK7FGUiISXg679o16BTJSs/V2bBOrX7cZGRGte
PMBGz19qX0rzoeNpLJFpJi23+/wAYaqzz0kyRGeyA0De0ugsqw2XRvUPciaYhbqqOraFUfmpyspC
o7ClzwsvnO1h1gVX/lDBwfLokrBeijsG3FN3Hg==</ds:SignatureValue>
                  <ds:KeyInfo>
                     <ds:X509Data>
                        <ds:X509Certificate>MIIFBDCCA+ygAwIBAgIkAhwR/
6b9CQnrHMhxuBSYqOCbHugRb+e4U/9jWi9kAgIWCzKcMA0GCSqG
SIb3DQEBBQUAMDExGjAYBgNVBAsTEU9yZ2FuaXphdGlvbmFsIENBMRMwEQYDVQQKFApuYW1zYl90
cmVlMB4XDTE0MDUyMTE4NTQwMVoXDTI0MDUyMTE4NTQwMVowHzEdMBsGA1UEAxMUbmFtc2IuYmxy
Lm5vdmVsbC5jb20wggEiMA0GCSqGSIb3DQEBAQUAA4IBDwAwggEKAoIBAQDRTsdCFBM3ImpIyRAj
OdFFEYbC/ykQUEZFwGp62BAUxoLIOpmDZyxpbqIh+l462GFByuCvkLOhnelOGV6Ii/cTAbaHko7h
T7cfUC3N4kmhnc3IXWgjodRIXMlaUSYDYd79guyVjG0brOWJMxJxvm1eo3p8bFzPLnpkEdJ7c8HM
BRqckecaGT8nbpm1KGZFAStrRRTryu2aG670FP3+MHWZmydqLlvrK1NCfe+7DlpOUwA13sSgMslf
6UCI4E50gn6pQ26rctGKrBsFfrX76t6ESZuaqFlWS+YA11cWS3irtihT0p2GsoxcJzq+IvHosHY+
pvrt4gcJiZJN6P3e6yrrAgMBAAGjggIUMIICEDAdBgNVHQ4EFgQUpSkUiviZFQ7yIDLb9sJT+mZH
kngwHwYDVR0jBBgwFoAUfFLP7EF6tU2u2qquPNTvLDdV7e8wggHMBgtghkgBhvg3AQkEAQSCAbsw
ggG3BAIBAAEB/xMdTm92ZWxsIFNlY3VyaXR5IEF0dHJpYnV0ZSh0bSkWQ2h0dHA6Ly9kZXZlbG9w
ZXIubm92ZWxsLmNvbS9yZXBvc2l0b3J5L2F0dHJpYnV0ZXMvY2VydGF0dHJzX3YxMC5odG0wggFI
oBoBAQAwCDAGAgEBAgFGMAgwBgIBAQIBCgIBaaEaAQEAMAgwBgIBAQIBADAIMAYCAQECAQACAQCi
BgIBFwEB/6OCAQSgWAIBAgICAP8CAQADDQCAAAAAAAAAAAAAAAADCQCAAAAAAAAAADAYMBACAQAC
CH//////////AQEAAgQG8N9IMBgwEAIBAAIIf/////////8BAQACBAbw30ihWAIBAgICAP8CAQAD
DQBAAAAAAAAAAAAAAAADCQBAAAAAAAAAADAYMBACAQACCH//////////AQEAAgQR/6b9MBgwEAIB
AAIIf/////////8BAQACBBH/pv2iTjBMAgECAgEAAgIA/wMNAIAAAAAAAAAAAAAAAMJAIAAAAAA
AAAAMBIwEAIBAAIIf/////////8BAQAwEjAQAgEAAgh//////////wEBADANBgkqhkiG9w0BAQUF
AAOCAQEAbA0AdHm5pV6cEwSyOoB3aJfaLegMYPlAuTNK9ajhez9PIHPGSQzNxTRbj3eV9P+ueP7j
i8AFVR3Ej4eA7S1i5kPGuSXhwM6VhSIsCn+x+HbpnFdWJdu5EvErjTIbbjRU/4wTRCqKe7loFqKs
rH+BGNuUJw16l2PM+wJ+sajX7ktzP8rk8CF+cTOe8ggFcEuJ4igl1MMkVbul1RTggRmpcILNFk57
QdmySozjVok1OVQOzIGcAggPBSZeCumNNP8mQIAmvnwWG0cTvDIkMkCV1AzCC0WK0dWM53JZD/aa
tHay9w8QWoUU5cJo8B+uSm2vN+53PdtMKWOXhJcXtXpKKg==</ds:X509Certificate>
                        </ds:X509Data>
                     </ds:KeyInfo>
                  </ds:Signature>
                  <saml2:Subject>
                     <saml2:NameID NameQualifier="">admin</saml2:NameID>
                     <saml2:SubjectConfirmation
Method="urn:oasis:names:tc:SAML:2.0:cm:bearer"/>
                  </saml2:Subject>
                  <saml2:Conditions NotBefore="2014-05-26T10:33:50.564Z"
```

```
NotOnOrAfter="2014-05-26T10:35:50.564Z">
                  <saml2:AudienceRestriction>
                     <saml2:Audience>http://164.99.184.228:8080/doubleit/services/
doubleit</saml2:Audience>
                  </saml2:AudienceRestriction>
               </saml2:Conditions>
               <saml2:Advice/>
               <saml2:AuthnStatement AuthnInstant="2014-05-26T10:33:50.564Z">
                  <saml2:AuthnContext>

<saml2:AuthnContextClassRef>urn:oasis:names:tc:SAML:1.0:am:password</
saml2:AuthnContextClassRef>
                  </saml2:AuthnContext>
               </saml2:AuthnStatement>
               <saml2:AttributeStatement>
                  <saml2:Attribute AttributeName="emailaddress"
AttributeNamespace="http://schemas.xmlsoap.org/ws/2005/05/identity/claims"
Name="emailaddress" NameFormat="http://schemas.xmlsoap.org/ws/2005/05/identity/
claims">
                     <saml2:AttributeValue>admin@idp.com</saml2:AttributeValue>
                  </saml2:Attribute>
               </saml2:AttributeStatement>
            </saml2:Assertion>
         </wst:RenewTarget>
      </wst:RequestSecurityToken>
     <ns:RequestSecurityToken xmlns:ns="http://docs.oasis-open.org/ws-sx/ws-trust/
200512/"/>
   </soap:Body>
</soap:Envelope>
```

Renew Token - Sample Response

```
<S:Envelope xmlns:S="http://www.w3.org/2003/05/soap-envelope" xmlns:wsse11="http:/
/docs.oasis-open.org/wss/oasis-wss-wssecurity-secext-1.1.xsd" xmlns:wsse="http://
docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd"
xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-
utility-1.0.xsd" xmlns:xs="http://www.w3.org/2001/XMLSchema">
   <S:Header>
      <Action S:mustUnderstand="true" xmlns="http://www.w3.org/2005/08/
addressing">http://docs.oasis-open.org/ws-sx/ws-trust/200512/RSTR/RenewFinal</
Action>
      <MessageID xmlns="http://www.w3.org/2005/08/addressing">uuid:f41d7aeb-6f67-
4df3-9fe4-e160889b7efb</MessageID>
      <RelatesTo xmlns="http://www.w3.org/2005/08/addressing">urn:uuid:9cfedcee-
2ebf-47e0-a24a-45281d785136</RelatesTo>
      <To xmlns="http://www.w3.org/2005/08/addressing">http://www.w3.org/2005/08/
addressing/anonymous</To>
      <wsse:Security S:mustUnderstand="true">
         <wsu:Timestamp wsu:Id="_1" xmlns:ns15="http://docs.oasis-open.org/ws-sx/
ws-secureconversation/200512" xmlns:ns14="http://schemas.xmlsoap.org/soap/
envelope/">
            <wsu:Created>2014-05-26T10:35:41Z</wsu:Created>
            <wsu:Expires>2014-05-26T10:40:41Z</wsu:Expires>
         </wsu:Timestamp>
      </wsse:Security>
   </S:Header>
   <S:Body>
      <trust:RequestSecurityTokenResponse xmlns:ns10="http://www.w3.org/2000/09/
xmldsig#" xmlns:ns13="http://www.w3.org/2001/10/xml-exc-c14n#" xmlns:ns3="http://
docs.oasis-open.org/wss/oasis-wss-wssecurity-secext-1.1.xsd" xmlns:ns5="http://
```

```
docs.oasis-open.org/ws-sx/ws-trust/200512/" xmlns:ns9="http://schemas.xmlsoap.org/
ws/2006/02/addressingidentity" xmlns:sc="http://docs.oasis-open.org/ws-sx/ws-
secureconversation/200512" xmlns:trust="http://docs.oasis-open.org/ws-sx/ws-trust/
200512" xmlns:wsa="http://www.w3.org/2005/08/addressing" xmlns:wsp="http://
schemas.xmlsoap.org/ws/2004/09/policy" xmlns:wst="http://schemas.xmlsoap.org/ws/
2005/02/trust">
         <trust:TokenType>urn:oasis:names:tc:SAML:2.0:assertion</trust:TokenType>
         <trust:RequestedSecurityToken>
            <saml2:Assertion ID="nstsa657b5f4-9bf0-45b7-9875-07eeb6d65196"
IssueInstant="2014-05-26T10:35:41.072Z" Version="2.0" xmlns:ds="http://www.w3.org/
2000/09/xmldsig#" xmlns:exc14n="http://www.w3.org/2001/10/xml-exc-c14n#"
xmlns:saml2="urn:oasis:names:tc:SAML:2.0:assertion" xmlns:xenc="http://www.w3.org/
2001/04/xmlenc#">
               <saml2:Issuer>https://namsb.blr.novell.com/nidp/wstrust/sts</
saml2:Issuer>
               <ds:Signature>
                  <ds:SignedInfo>
                     <ds:CanonicalizationMethod Algorithm="http://www.w3.org/2001/
10/xml-exc-c14n#"/>
                     <ds:SignatureMethod Algorithm="http://www.w3.org/2000/09/
xmldsig#rsa-sha1"/>
                    <ds:Reference URI="#nstsa657b5f4-9bf0-45b7-9875-07eeb6d65196">
                        <ds:Transforms>
                           <ds:Transform Algorithm="http://www.w3.org/2000/09/
xmldsig#enveloped-signature"/>
                           <ds:Transform Algorithm="http://www.w3.org/2001/10/xml-
exc-c14n#"/>
                        </ds:Transforms>
                        <ds:DigestMethod Algorithm="http://www.w3.org/2000/09/
xmldsig#sha1"/>
                        <ds:DigestValue>bX5LSfro0HkupLsMkU/V+x39P+g=</
ds:DigestValue>
                     </ds:Reference>
                  </ds:SignedInfo>
                  <ds:SignatureValue>QLRRXQ4TzTgM9mVa5UF1p7YRqRvLP/
h3pyP0KVzZXcbCfDmtT4bO14lqfhNoXL+Ym2iu2V1MIC5I
TRSt6D/y6pfs4/nChMrOuk5spMZYLBee+0PdlGYhfLGzyh/AONZGsoVrHf1/LItMeTp4MVmk/hTp
8yTWb0r79Ssz5TEbwJ/NkqFXxa9XffheaTySOfNXQYu3tL1rdp7Zaq5BR7mye00huo6gBTshHTXM
fGPYMu/Sy0kapqTBWHUbwt8FzysBEgELZdquhuvt1NOfHqkWAbyp5vExjjrYxl06Z7Fu3LnDSq+m
hI9S+VLs1bBR2XgNofhw/bFVBboYkzZDT6Ipmg==</ds:SignatureValue>
                  <ds:KeyInfo>
                     <ds:X509Data>
                        <ds:X509Certificate>MIIFBDCCA+ygAwIBAgIkAhwR/
6b9CQnrHMhxuBSYqOCbHugRb+e4U/9jWi9kAgIWCzKcMA0GCSqG
SIb3DQEBBQUAMDExGjAYBgNVBAsTEU9yZ2FuaXphdGlvbmFsIENBMRMwEQYDVQQKFApuYW1zYl90
cmVlMB4XDTE0MDUyMTE4NTQwMVoXDTI0MDUyMTE4NTQwMVowHzEdMBsGA1UEAxMUbmFtc2IuYmxy
Lm5vdmVsbC5jb20wggEiMA0GCSqGSIb3DQEBAQUAA4IBDwAwggEKAoIBAQDRTsdCFBM3ImpIyRAj
OdFFEYbC/ykQUEZFwGp62BAUxoLIOpmDZyxpbqIh+l462GFByuCvkLOhnelOGV6Ii/cTAbaHko7h
T7cfUC3N4kmhnc3IXWgjodRIXMlaUSYDYd79guyVjG0brOWJMxJxvm1eo3p8bFzPLnpkEdJ7c8HM
BRqckecaGT8nbpm1KGZFAStrRRTryu2aG670FP3+MHWZmydqLlvrK1NCfe+7DlpOUwA13sSgMslf
6UCI4E50gn6pQ26rctGKrBsFfrX76t6ESZuaqFlWS+YA11cWS3irtihT0p2GsoxcJzq+IvHosHY+
pvrt4gcJiZJN6P3e6yrrAgMBAAGjggIUMIICEDAdBgNVHQ4EFgQUpSkUiviZFQ7yIDLb9sJT+mZH
kngwHwYDVR0jBBgwFoAUfFLP7EF6tU2u2qquPNTvLDdV7e8wggHMBgtghkgBhvg3AQkEAQSCAbsw
ggG3BAIBAAEB/xMdTm92ZWxsIFNlY3VyaXR5IEF0dHJpYnV0ZSh0bSkWQ2h0dHA6Ly9kZXZlbG9w
ZXIubm92ZWxsLmNvbS9yZXBvc2l0b3J5L2F0dHJpYnV0ZXMvY2VydGF0dHJzX3YxMC5odG0wggFI
oBoBAQAwCDAGAgEBAgFGMAgwBgIBAQIBCgIBaaEaAQEAMAgwBgIBAQIBADAIMAYCAQECAQACAQCi
BgIBFwEB/6OCAQSgWAIBAgICAP8CAQADDQCAAAAAAAAAAAAAAADCQCAAAAAAAAAADAYMBACAQAC
CH/////////AQEAAgQG8N9IMBgwEAIBAAIIf////////8BAQACBAbw30ihWAIBAgICAP8CAQAD
DQBAAAAAAAAAAAAAAADCQBAAAAAAAAAADAYMBACAQACCH/////////AQEAAgQR/6b9MBgwEAIB
AAIIf////////8BAQACBBH/pv2iTjBMAgECAgEAAgIA/wMNAIAAAAAAAAAAAAAAAMJAIAAAAAA
```

```
AAAAMBIwEAIBAAIIf/////////8BAQAwEjAQAgEAAgh/////////wEBADANBgkqhkiG9w0BAQUF
AAOCAQEAbA0AdHm5pV6cEwSyOoB3aJfaLegMYPlAuTNK9ajhez9PIHPGSQzNxTRbj3eV9P+ueP7j
i8AFVR3Ej4eA7S1i5kPGuSXhwM6VhSIsCn+x+HbpnFdWJdu5EvErjTIbbjRU/4wTRCqKe7loFqKs
rH+BGNuUJw16l2PM+wJ+sajX7ktzP8rk8CF+cTOe8ggFcEuJ4igl1MMkVbul1RTggRmpcILNFk57
QdmySozjVok1OVQOzIGcAggPBSZeCumNNP8mQIAmvnwWG0cTvDIkMkCV1AzCC0WK0dWM53JZD/aa
tHay9w8QWoUU5cJo8B+uSm2vN+53PdtMKWOXhJcXtXpKKg==</ds:X509Certificate>
                     </ds:X509Data>
                  </ds:KeyInfo>
               </ds:Signature>
               <saml2:Subject>
                  <saml2:NameID NameQualifier="">admin</saml2:NameID>
                  <saml2:SubjectConfirmation
Method="urn:oasis:names:tc:SAML:2.0:cm:bearer"/>
               </saml2:Subject>
               <saml2:Conditions NotBefore="2014-05-26T10:35:41.072Z"
NotOnOrAfter="2014-05-26T10:37:41.072Z">
                  <saml2:AudienceRestriction>
                     <saml2:Audience>http://164.99.184.228:8080/doubleit/services/
doubleit</saml2:Audience>
                  </saml2:AudienceRestriction>
               </saml2:Conditions>
               <saml2:Advice/>
               <saml2:AuthnStatement AuthnInstant="2014-05-26T10:33:50.564Z">
                  <saml2:AuthnContext>

<saml2:AuthnContextClassRef>urn:oasis:names:tc:SAML:1.0:am:password</
saml2:AuthnContextClassRef>
                  </saml2:AuthnContext>
               </saml2:AuthnStatement>
               <saml2:AttributeStatement>
                  <saml2:Attribute AttributeName="emailaddress"
AttributeNamespace="http://schemas.xmlsoap.org/ws/2005/05/identity/claims"
Name="emailaddress" NameFormat="http://schemas.xmlsoap.org/ws/2005/05/identity/
claims">
                     <saml2:AttributeValue xmlns:soap="http://www.w3.org/2003/05/
soap-envelope">admin@idp.com</saml2:AttributeValue>
                  </saml2:Attribute>
               </saml2:AttributeStatement>
            </saml2:Assertion>
         </trust:RequestedSecurityToken>
         <trust:Lifetime>
            <wsu:Created>2014-05-26T10:35:41.071Z</wsu:Created>
            <wsu:Expires>2014-05-26T10:37:41.071Z</wsu:Expires>
         </trust:Lifetime>
      </trust:RequestSecurityTokenResponse>
   </S:Body>
</S:Envelope>
```

Configuring OAuth and OpenID Connect

OAuth 2.0 is an open protocol to allow secure authorization. It enables users to allow third-party client to access users' private resources. Users do not need to share their credentials. The third-party clients can be web applications, mobile phones, handheld devices, and desktop applications. OAuth provides a method to issue Access tokens to third-party client applications with the user's approval. The third-party client application can then use the Access token to access protected resources hosted by the resource server.

OAuth allows users to control the access to web resources by scope, action, and time.

Access Manager uses OpenID Connect along with OAuth. OpenID Connect implements a single sign-on protocol on top of the OAuth authorization process. It allows client applications to verify the identity of a user based on the authentication performed by the Identity Server (authorization server). It also allows client applications to obtain a user's basic profile information.

See Appendix C, "OAuth versus Other Protocols," on page 1195 for differences among OAuth and other single sign-on protocols.

- "How OAuth Helps" on page 533
- "OAuth Terminology" on page 533
- "How OAuth Works" on page 534
- "Why Use OpenID Connect" on page 536
- "OAuth Scenarios" on page 537
- "OAuth Implementation Flow" on page 539
- "OAuth Authorization Grant" on page 542
- "OpenID Connect Authentication Flows" on page 544
- "Managing OAuth and OpenID Connect" on page 545
- "Configuring the Access Gateway for OAuth" on page 553
- "Viewing Endpoint Details" on page 556
- "OAuth and OpenID Connect Audit Events" on page 557
- "Enabling Logging for OAuth and OpenID Connect" on page 557
- "Managing Clients Applications by Using REST API" on page 557
- "Managing OAuth 2.0 Resource Server and Scope by Using REST API" on page 560
- "End User Operations" on page 563
- "Configuring the Demo OAuth Application" on page 564

How OAuth Helps

OAuth addresses the following security concerns:

- To provide access to protected resources, users share their credentials in clear-text with third-party applications. Potential security breaches that can result from the ability of third-party applications to store a user's credentials for future use.
- Security issues associated with password authentication.
- The inability of resource owners to restrict a client application's access to protected resources for a specified duration or to limit the client application's access to a subset of resources.
- The inability of resource owners to revoke a client application's access to a specific client application.

OAuth Terminology

The following list includes frequently used terms in an OAuth flow:

OAuth Roles

- **Resource Owner:** Grants access to a protected resource.
- **Resource Server:** Hosts the protected resources. It accepts and responds to requests by using Access tokens.

- **Client:** An application that requests access to protected resources on behalf of the resource owner with the resource owner's authorization. A client application, for example, can be a gaming application.
- **Authorization Server:** Generates Access tokens for a client application after authenticating the resource owner and obtaining authorization from the resource owner. The authorization server in Access Manager is the Identity Server.

OAuth Credentials and Tokens

- **ID Token: JSON Web Token (JWT)**: Contains a user's claims such as identity, email address, and other profile information. It also specifies the issuing authority.
- **Access Token:** Required to access protected resources. Contains the attributes, such as scope and duration, that are granted by the authorization server.
- **Refresh Token:** Used to obtain Access tokens. The authorization server issues a Refresh token to the client application. Client applications use this token to obtain a new Access token when the current Access token expires or is no longer valid.
- **Client Key and Secret:** A client application uses a client key to identify itself to a service provider. A client application uses the client secret to establish the ownership of the client key. The authorization server assigns a key and a secret to a client application while registering it.

OAuth Endpoints

- **Authorization Endpoint:** Client applications use this endpoint to interact with the resource owner and obtain an authorization grant. It is located on an authorization server.
- **Token Endpoint:** Client applications use this endpoint to obtain an Access token by providing their authorization grant or Refresh token. It is also located on an authorization server.

How OAuth Works

You can configure OAuth with Access Manager in one of two ways based on the following requirements:

- If the web applications validate the Access token before allowing a client application accesses to resources.
- If the Access Gateway validates the Access token on behalf of web applications.

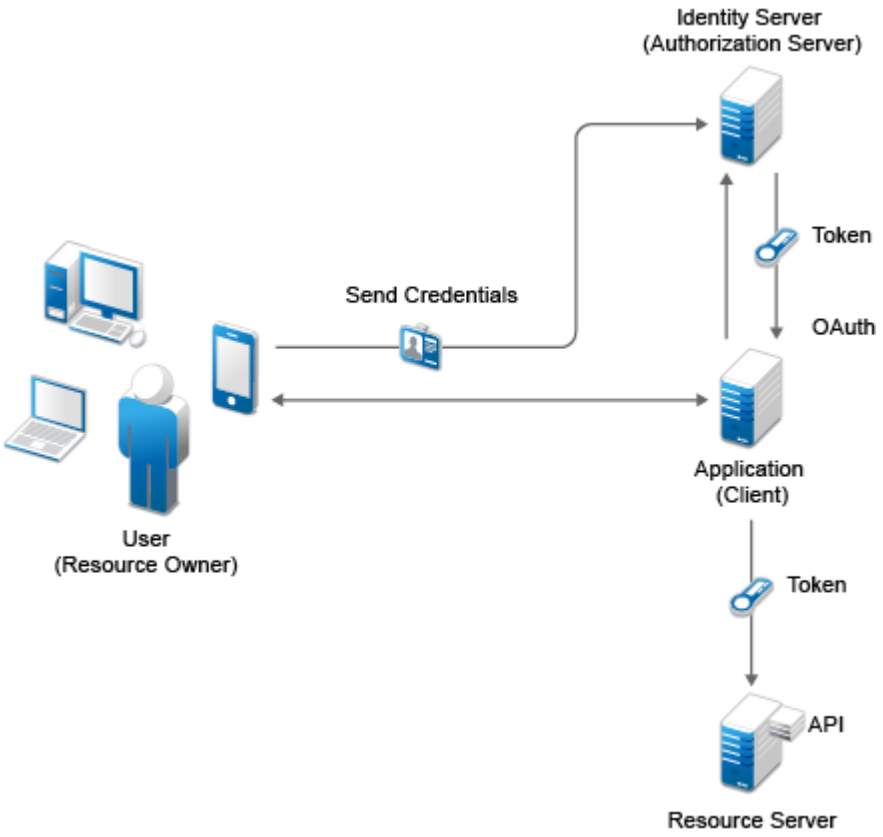
Web applications validate an Access token before allowing a client application to access resources

The Identity Server (authorization server) issues an Access token and web applications validate the token before granting a client application to access resources. This configuration is suitable in the following scenarios:

- Web server authentication
- Accessing resources without using owner's credentials
- RESTful applications security
- Mobile authentication

For more information about these scenarios, see "OAuth Scenarios" on page 537.

Figure 5-26 *The following diagram illustrates the OAuth flow:*

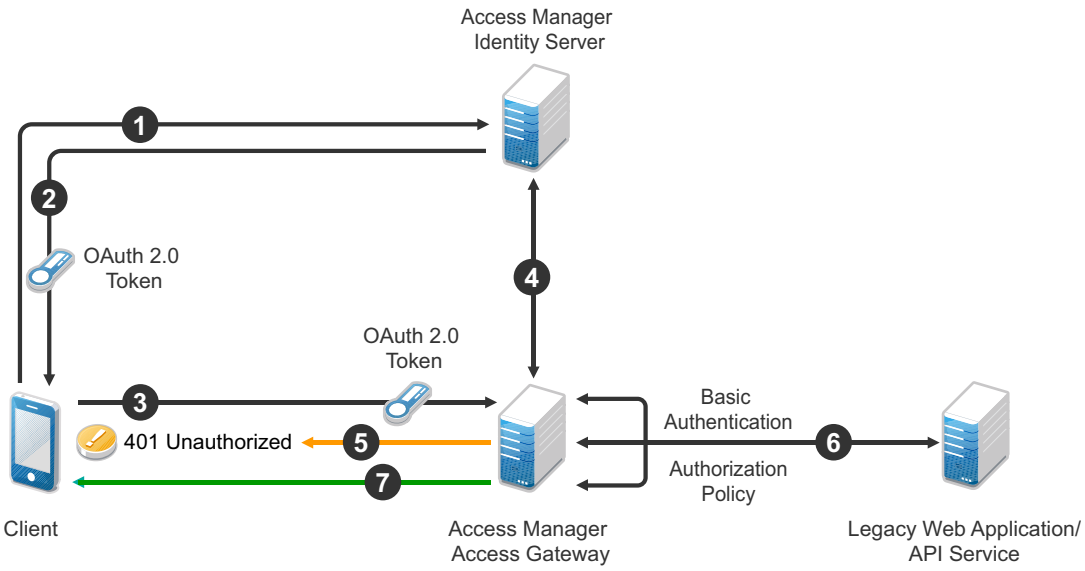


1. The client application requests authorization from the user (resource owner). Client applications can make the authorization request directly to the resource owner or through the authorization server (Identity Server) as an intermediary.
2. The client application receives an authorization grant from the authorization server. An authorization grant represents the resource owner's authorization. The user communicates the authorization by using one of four grants types (see “OAuth Authorization Grant” on page 542) or by using an extension grant.
3. The client application authenticates itself at the authorization server, sends the authorization grant, and requests an Access token.
4. The authorization server authenticates the client application and validates the authorization grant. The authorization server issues an Access token for a valid grant.
5. The client application requests the resource server to provide access to the protected resource and authenticates this by presenting the Access token.
6. The resource server accepts the request for a valid token.

The Access Gateway validates the Access token on behalf of web applications

This configuration is suitable when client applications want to access resources on legacy web applications. See “Legacy Web Applications Security” on page 539 for details about when to use this configuration.

Figure 5-27 *The following diagram illustrates the workflow:*



1. A client application requests access to a web resource and provides authentication details to the Identity Server.
2. The Identity Server authenticates the client application, gets the user's consent, generates an OAuth token, and sends the token to the client application.
3. The client application provides the token to the Access Gateway.
4. The Access Gateway sends the token to the Identity Server for validation.
5. If the token is not valid, AG returns a 401 error.
6. If the token is valid, The Access Gateway performs the following tasks:
   - **6a** Executes the authorization policy, if configured, based on OAuth scopes or claims.
   - **6b** Sends user attributes and grants details provided to the client application to the web application by using the Identity Injection policy, if configured.
7. The resource server returns a response to the Access Gateway and the Access Gateway sends this response to the client application.

Why Use OpenID Connect

OAuth allows users to authorize client applications to access users' protected resources through an Access token. The Access token does not contain any information about a user's identity. Hence, a client application does not know who the user is. A client application also does not know if the authorization server has issued the access token to it or to any other relying party.

OpenID Connect builds on OAuth and provides solutions for OAuth's limitations. It issues an ID token that contains signed assertions about the user. Client applications can verify the ID token and obtain additional details about the user. The ID token also contains information about the issuing authority, the intended client application, time of the token created, and the token expiration time.

OpenID Connect Claims

A client application can obtain information about a user and authentication events through claims. A claim contains information about a user such as phone number, first name, and last name.

OAuth Scenarios

This section describes the following basic scenarios supported by OAuth:

- "RESTful Applications Security" on page 537
- "Accessing Resources without Using the Resource Owner's Credentials" on page 537
- "Web Server Authentication" on page 537
- "Mobile Authentication" on page 539
- "Legacy Web Applications Security" on page 539

RESTful Applications Security

OAuth provides a way to secure REST APIs. For example, an enterprise `acme.com` exposes REST APIs that provide various functions. Using OAuth, `acme.com` can provide secure authorization control on APIs to ensure that the right people have access to these APIs. In addition, they can also enable applications to call APIs on behalf of a user. `acme.com` can also revoke access to an API even if an application uses it.

See Appendix B, "SOAP versus REST API," on page 1193 to learn the differences between SOAP and REST API.

Accessing Resources without Using the Resource Owner's Credentials

In a typical web authentication model, a client application uses the resource owner's credentials to access the resource owner's information that is hosted on a server. OAuth allows a client application to access resources that are controlled by the resource owner and provides a method to obtain permission from the resource owners to access their resources. The resource owners provide this permission in the form of a token and a matching shared-secret. The resource owner does not need to share credentials with the client application. Tokens are issued with a restricted scope and limited life.

For example, a user named Alice has installed a gaming application that runs in her browser and uses OAuth for accessing a social site at `www.example.com`. The gaming application updates scores in a database at `www.example.com`. The gaming application is registered with the social site and has an identifier. Alice has registered with the social site for identification and authentication. To upload Alice's scores, the gaming application accesses the score database when Alice authorizes it. When Alice accesses the page from the redirect URI in the game application, the authorization serversends the client ID, password, and authentication code received in the redirect request parameters to `www.example.com`, which in turn returns an Access token to the game application. The gaming application sends the token to `www.example.com` to access Alice's resources. `www.example` verifies the token and grants the gaming application access to Alice's account for updating the scores.

NOTE: This example is derived from the OAuth RFC document.

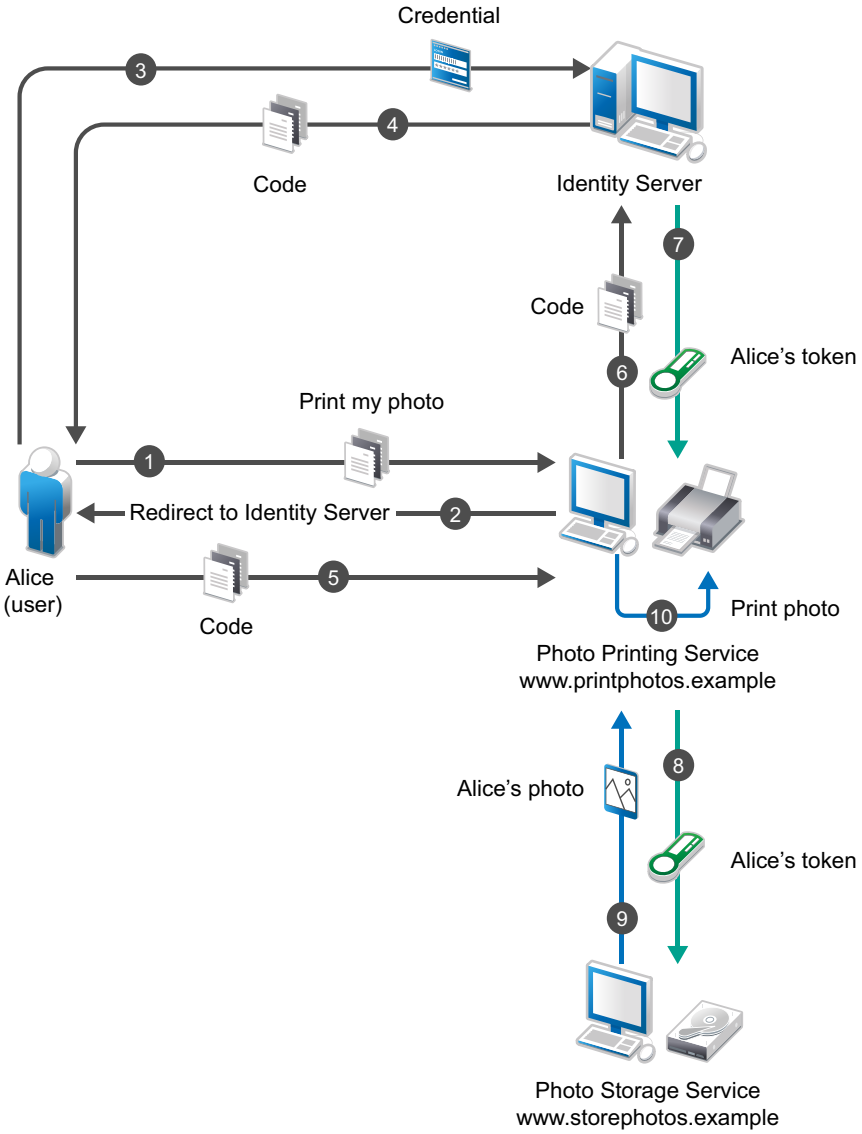
Web Server Authentication

To access the user data of other sites, third-party applications seek user's credentials (user names and passwords). Sharing passwords may result in a security breach. OAuth enables third-party applications to access a user's information without using the users' passwords. OAuth also provides a way to grant limited access. For example, a user (resource owner) can allow a printing service

(client application) to access private photos stored at a photo sharing service (server), without sharing credentials with the printing service. Instead, the user authenticates directly with the photo sharing service that issues the printing service delegation-specific credentials.

For example, a user named Alice accesses an application running on a web server at `www.printphotos.example` and instructs it to print her photographs that are stored on a server at `www.storephotos.example`. The application at `www.printphotos.example` receives Alice's authorization consent for accessing her photographs without learning her authentication credentials of `www.storephotos.example`.

The following diagram illustrates the workflow of the web server authentication:



NOTE: This example is derived from the OAuth RFC document.

Mobile Authentication

Applications on a mobile device request for authentication and the web server redirects you to the authorization server (Identity Server) to authenticate and authorize the server to access your data. When you approve, the web server receives an Access token as part of the redirect URL. After the authorization server grants the token, the application can access the protected data with the Access token. Less confidential applications, such as mobile clients or thick clients use this authentication.

Legacy Web Applications Security

OAuth enables client applications including thick clients, mobile, and OAuth enabled browser-based applications to access resources available on legacy web applications. In this scenario, the Access Gateway validates the token on behalf of web applications. You do not need to modify web applications to understand the OAuth flow.

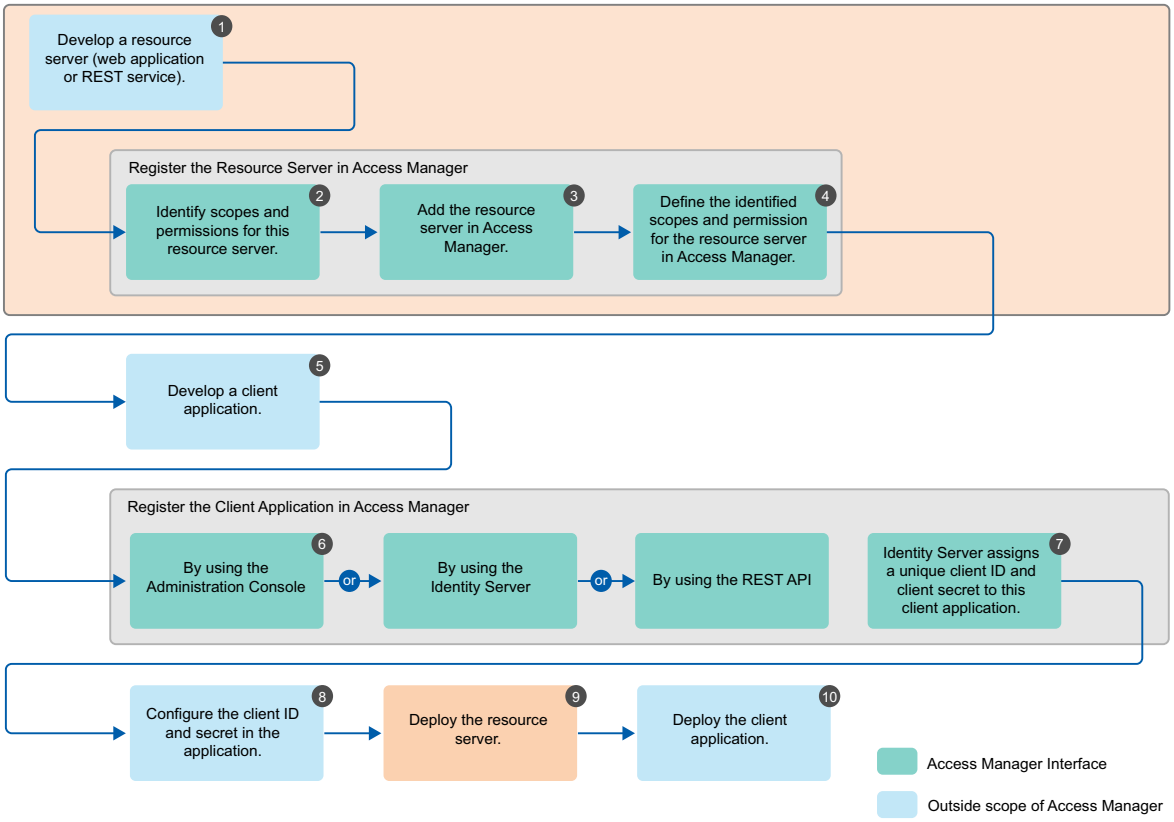
OAuth Implementation Flow

The following diagrams depict the implementation flow of OAuth in Access Manager:

- OAuth Implementation without using the Access Gateway
- OAuth Implementation using the Access Gateway

NOTE: Access Manager uses variable length Access tokens and authorization codes. The client applications and web servers must not assume any fixed size of tokens and codes and must allocate necessary memory to handle the token. Token size depends on the size of scope names. Some servers may have size limitations on query string and HTTP headers. Ensure that an application uses only necessary scopes to avoid any issue.

OAuth Implementation without using the Access Gateway

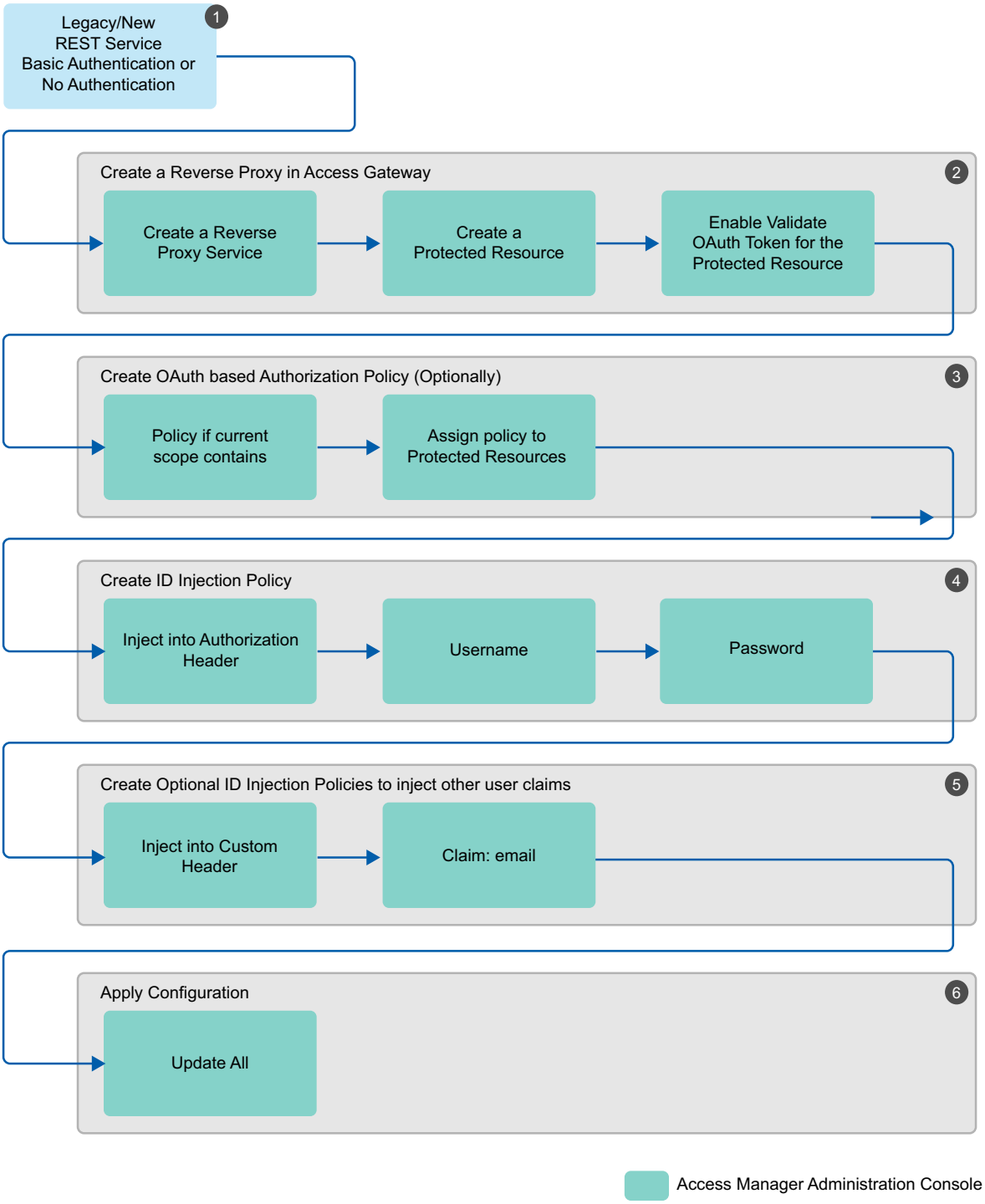


1. Develop a resource server (web application or REST service). (Application Developer)
2. Identify scopes and permissions for this resource server. (Application Developer or Administrator)
3. Add the resource server in Access Manager. See “Adding a Resource Server” on page 549. (Application Developer or Administrator)
4. Define the identified scopes and permissions for the resource server in Access Manager. See “Defining Scopes and Claims for a Resource Server” on page 549. (Application Developer or Administrator)
5. Develop a client application. (Application Developer)
6. Register the client application in Access Manager. You can register a client by using the Administration Console (Administrator), Identity Server (Application Developer or Administrator), or REST API (Application Developer or Administrator). See “Registering OAuth Client Applications” on page 551 and “Registering a Client Application by Using REST API” on page 558.
7. Identity Server assigns a unique client ID and client secret to this client application.
8. Configure the client ID and secret in the application. (Application Developer or Administrator)
9. Deploy the resource server. (Application Developer)
10. Deploy the client application. (Application Developer)

For information about basic scenarios in which you can implement this configuration, see “OAuth Scenarios” on page 537.

For more information about how to enable and configure OAuth in Access Manager for this implementation flow, see “Managing OAuth and OpenID Connect” on page 545.

OAuth Implementation using the Access Gateway



1. Determine the web application or REST service for which you want to implement this configuration.

2. Create a reverse proxy in the Access Gateway and enable OAuth in the Access Gateway for this reverse proxy. See “Enabling OAuth in the Access Gateway” on page 553.
3. Configure an authorization policy based on OAuth Scopes. See “Configuring an Authorization Policy based on OAuth Scopes” on page 554.
4. Configure an Identity Injection policy to inject user name and password. See “Configuring an Identity Injection Policy for OAuth Claims” on page 555.
5. Configure optional Identity Injection policies to inject other user claims, if required. You can define the additional roles in the same policy also that you configured for injecting user name and password. See “Configuring an Identity Injection Policy for OAuth Claims” on page 555.
6. Apply the changes.

For information about the scenario in which you can implement this configuration, see “Legacy Web Applications Security” on page 539.

For information about how to configure OAuth in Access Manager for this implementation flow, see “Configuring the Access Gateway for OAuth” on page 553.

OAuth Authorization Grant

Authorization grant is an intermediate credential that represents the resource owner authorization. To request an Access token, the client application obtains authorization from the resource owner. The resource owner communicates the authorization in the form of an authorization grant that a client application uses to request the Access token. OAuth defines four grant types: authorization code, implicit, resource owner credentials, and client credentials. It also provides an extension mechanism for defining additional grant types. This section discusses the authorization flow for the following grants:

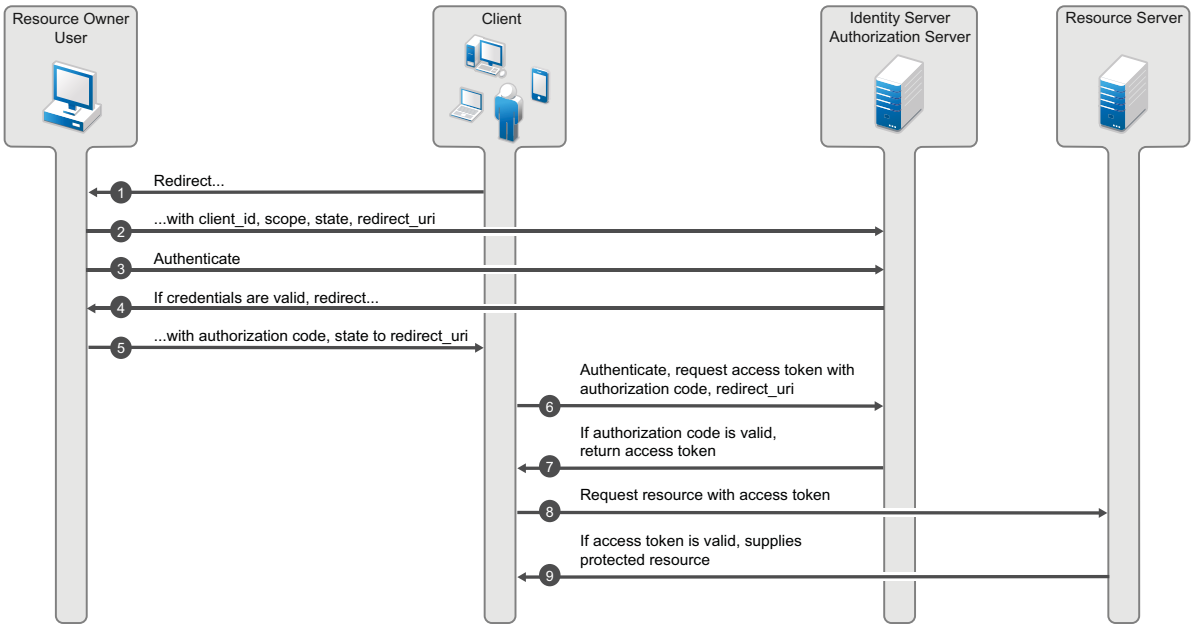
- “Authorization Code Grant (Web Server)” on page 542
- “Implicit Grant” on page 543
- “Resource Owner Credential Grant” on page 544
- “Client Credential Grant” on page 544

Authorization Code Grant (Web Server)

Client applications hosted on a secure server use Authorization Code Grant. Client applications use this grant to obtain both Access tokens and Refresh tokens. This grant ensures that both types of tokens remain with the client web application (the server side) and the authorization server does not send these to the browser. Only the authorization code is visible to the browser.

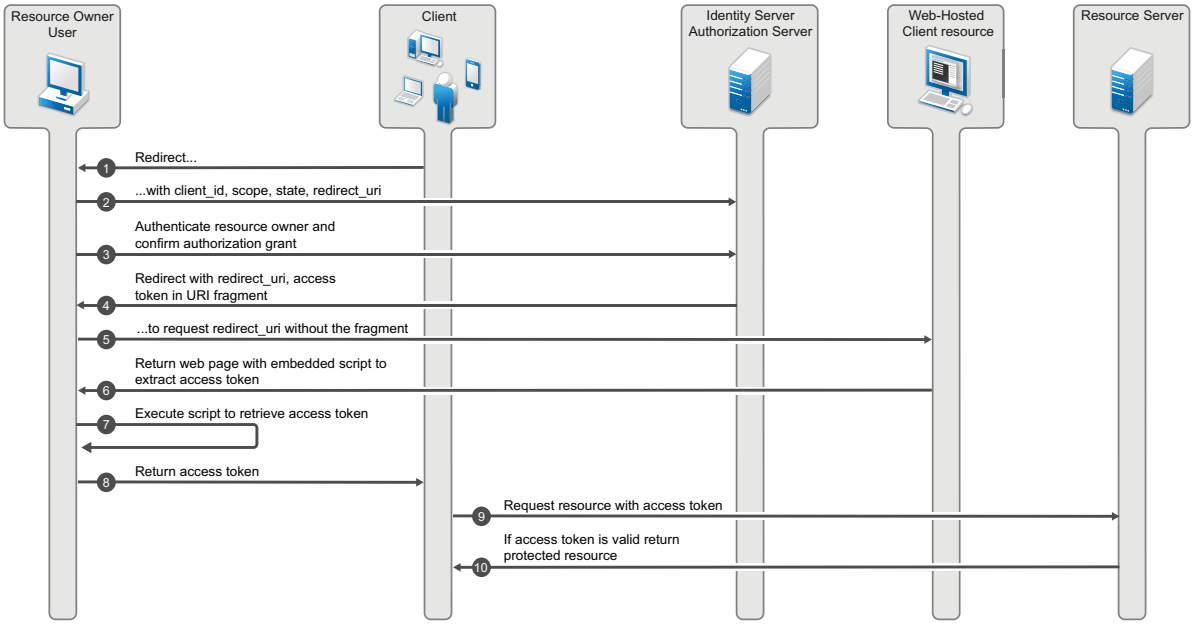
The client application redirects the resource owner to the authorization server through the web browser. The resource owner authenticates at the authorization server. The authorization server obtains resource owner’s consent and then redirects the web browser with the authorization code to the client application.

This flow is suitable for client applications who can interact with the resource owner's user-agent and can receive incoming requests from the authorization server.



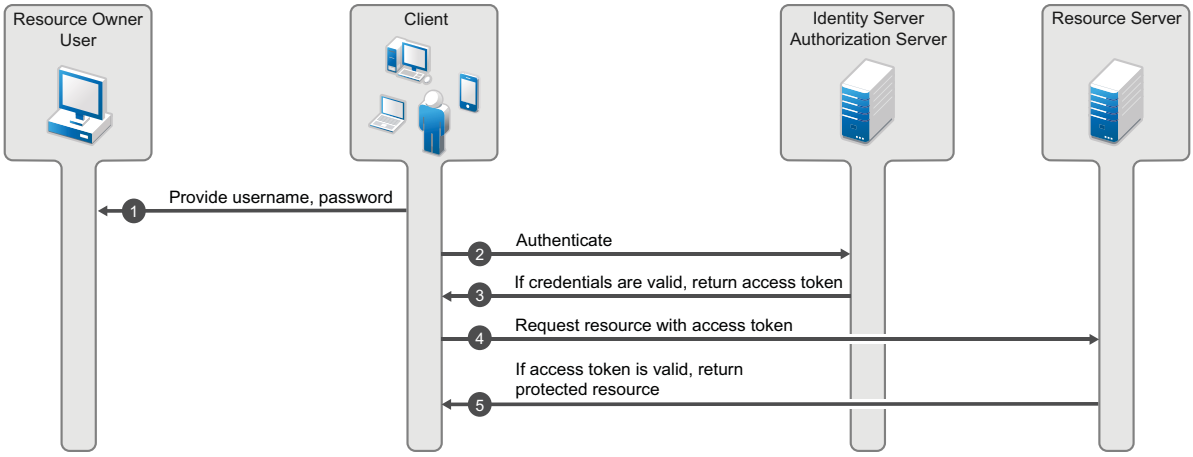
Implicit Grant

This flow is suitable for client applications residing in the user's device. A client application can implement this flow in a browser using a scripting language such as JavaScript or Flash, from a mobile device, or from a desktop application. After a user grants the requested authorization, the authorization server returns an Access token to the application. An intermediate authorization code is not required. As the authorization server sends the Access token to the web browser, this flow offers less security than the authorization code.



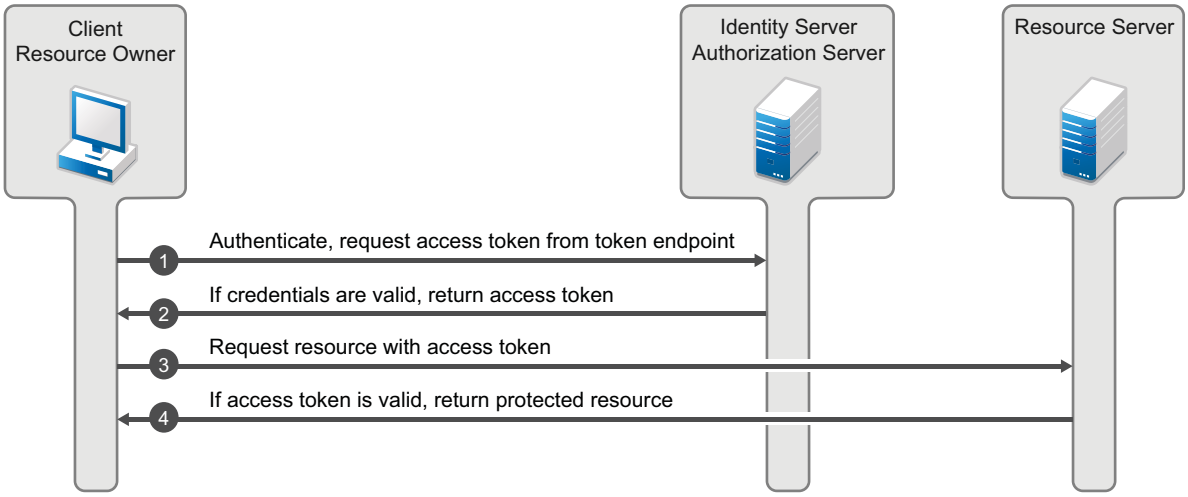
Resource Owner Credential Grant

This flow is suitable for client applications who have a trust relationship with resource owners.In this flow, the client application sends user's credentials along with its own credentials to the authorization server (Identity Server). The Identity Server provides an Access token and an Refresh token to the client application. The user does not need to log in to approve the request.



Client Credential Grant

The Client Credential Grant is useful for applications that access their own resources from the resource server. This grant type only requires the client application’s credentials. Resource owner's credentials are not required.



OpenID Connect Authentication Flows

This section describes the flow of OpenID Connect authentication by using the Authorization Code flow and Implicit flow.

- “Authentication by Using the Authorization Code Flow” on page 545
- “Authentication by Using the Implicit Flow” on page 545

Authentication by Using the Authorization Code Flow

In this authentication process, the Token endpoint returns all tokens.The Authorization Code Flow returns an authorization code to the client application. The client application exchanges it for an ID token and an Access token. The authorization server can also authenticate the client application before exchanging the authorization code for an Access token. This process does not expose tokens to the User Agent.

Process Flow:

1. The client application prepares an authentication request containing the desired request parameters and sends the request to the authorization server.
2. The authorization server authenticates the user.
3. The authorization server obtains the user consent for the request.
4. The authorization server sends the user consent to the client application with an authorization code.
5. The client application requests a response by using the authorization code at the Token endpoint.
6. The client application receives a response that contains an ID token and Access token in the response body.
7. The client application validates the ID token and retrieves the user's subject identifier.

Authentication by Using the Implicit Flow

In this authentication process, the authorization endpoint returns all tokens. The endpoint returns the Access token and ID token directly to the client application that may result in revealing the tokens to the user and applications that have access to the User Agent.

Process Flow:

1. The client application generates an authentication request containing the desired request parameters and sends the request to the authorization server.
2. The authorization server authenticates the user.
3. The authorization server obtains the user consent.
4. The authorization server sends the user to the client application with an ID token and, if requested, an Access token.
5. The client application validates the ID token and retrieves the user's Subject Identifier.

Managing OAuth and OpenID Connect

NOTE: NTS will support the Access Manager setup and any app issues where the API request is sent to the right Access Manager endpoint. Any other code changes needed to integrate with Access Manager are outside the scope of traditional NTS support and need to go through the namsdk@netiq.com channel.

The following is the sequence of the OAuth and OpenID Connect configuration:

1. Enable the OAuth protocol in the Administration Console
2. Define the global settings
3. Configure a resource server

4. Configure scopes and claims for a resource server
5. Register client applications

NOTE: Use Internet Explorer 10 or later, Firefox, or Chrome for configuring OAuth 2.0.

This section discusses the following topics:

- "Extending a User Store for OAuth 2.0 Authorization Grant Information" on page 546
- "Enabling OAuth and OpenID Connect" on page 547
- "Defining Global Settings" on page 547
- "Configuring a Resource Server" on page 548
- "Modifying Scopes of a Resource Server" on page 550
- "Modifying Claims and Attributes" on page 551
- "Managing OAuth Client Applications" on page 551

Extending a User Store for OAuth 2.0 Authorization Grant Information

Access Manager OAuth 2.0 implementation stores the information about a client application, which a user authorizes to access attributes and resources. This information is unique per user. So, you need to store as part of a User Object in the user store. If you already have a free attribute, you can use it in **Authorization Grant LDAP Attribute** in while defining Global Settings.

If a free attribute is not available, then extend the User Object schema to add a new single-valued *stream* attribute with a name. Access Manager will store an XML object in this attribute for each user authorization.

The following example describes how to extend the schema of a User Object in eDirectory:

1. Click to **Roles and Tasks** > **Schema** > **Create Attribute**.
2. Specify **Attribute Name** as `nidsOAuthGrant`.
3. Click **Next**.
4. Select **Stream** under **Syntax**.
5. Click **Next**.
6. Select **Single Valued**.
7. Click **Next** > **Finish**.
8. Go to **Roles and Tasks** > **Schema** > **Add Attribute**.
9. Select Person under **Available Classes**.
10. Click **OK**.
11. Move `nidsOAuthGrant` from **Available optional attributes** to **Optional attributes**.
12. Click **OK**.

Enabling OAuth and OpenID Connect

Access Manager ships with only SAML 1.1, Liberty, and SAML 2.0 enabled by default. To use OAuth, you must enable it in the Identity Server. Otherwise, the configuration will not work.

To enable OAuth and OpenID Connect, perform the following steps:

1. In the Administration Console, click **Devices** > **Identity Servers** > **Edit**.
2. In the **Enabled Protocols** section, select **OAuth & OpenID Connect**.
3. Click **OK**.
4. Update the Identity Server.

Defining Global Settings

The Global Settings page enables you to specify the default OAuth and OpenID Connect settings for the authorization server such as issuer URL, token types, grants, and so on.

1. In the Administration Console, click **Devices** > **Identity Server** > **Edit** > **OAuth & OpenID Connect** > **Global Settings**.
2. You can configure and view the following details on this page:

Field	Description
Issuer	Specify the name of the authorization server. This name is part of the ID token.
Authorization Grant LDAP Attribute	Specify a valid LDAP attribute. The attribute specified must exist in the user store. A scope uses this attribute to store user consent. Whenever a user grants authorization to an application, this attribute is updated. Ensure that no other application uses this attribute. For more information, see "Extending a User Store for OAuth 2.0 Authorization Grant Information" on page 546.
CORS Domains	Select any one of the following options based on your requirement: ◆ **None:** If you want to deny access for requests from all domains other than the domain of the resource. The resource referred here are resources such as Javascript on the client application. ◆ **Allow All:** If you want to allow access for requests from any domains. ◆ **Limit to:** If you want to allow access for requests from only selected domains. Specify the domain.

Field	Description
Grant Type(s)	Select the types of grants that the authorization server will support. Based on the grant type you select, the system selects corresponding token type by default. For more information about grant types, see “OAuth Authorization Grant” on page 542.
Token Type(s)	Select the types of tokens that the authorization server will support. ◆ **ID Token:** A security token that contains claims about the authentication of an end user by an authorization server to the relying party. ◆ **Access Token:** Includes the specific scopes and durations of granted access. ◆ **Refresh Token:** Used to obtain a new access token when an Access token becomes invalid or expires.
Authorization Code Timeout	Specify the duration in minute after how long the authorization code becomes invalid.
Access Token and ID Token Timeout	Specify the duration in minute after how long the Access token and ID token become invalid.
Refresh Token Timeout	Specify the duration in minute after how long the Refresh token becomes invalid.
Supports Signing	Select this option if you want the authorization server to sign the ID token If you selected **Supports Signing**, select a signing certificate.
Signing Certificate	Select a certificate to sign ID tokens.

3 Click **OK**.

Configuring a Resource Server

Access Manager allows you to define settings for each resource server. A resource server validates and accepts tokens sent by client applications, and then grants access to resources.

Setting up a resource server includes adding a resource server, defining scopes, and defining claims associated with each scope. Scopes and claims decide what resources client applications can access and what actions they can perform on the resources.

Access Manager also allows you to modify and delete configured resource servers. Configuring a resource server consists of the following actions:

- “Adding a Resource Server” on page 549
- “Defining Scopes and Claims for a Resource Server” on page 549
- “Restricting the Number of Requests” on page 550

Adding a Resource Server

Perform the following steps:

1. In the Administration Console, click **Devices** > **Identity Server** > **Edit** > **OAuth & OpenID Connect** > **Resource Server**.
2. Click **New**.
3. Specify a name for the resource server.
4. Click **OK**.

   Continue with "Defining Scopes and Claims for a Resource Server" on page 549.

Defining Scopes and Claims for a Resource Server

A scope is a set of permissible actions that a client application can perform on the accessed resources. You can define scopes and then set up claims or permissions for each scope. When a user grants client applications access to protected resources, they can perform actions based on permissions defined in the scope.

A scope can be of the following two types:

- Resource permissions: The resource server can fetch these scopes/permissions specific to requirements. For example, for the edit_photo scope, the resource server recognizes and grants edit rights for the photo.
- User Claims: Access Manager acts as a special resource server managing access to users' claims. A client application can request these claims.

For example, you can define a scope named email and define permissions associated with this scope such as read only. A client application who will access the email can only read the content.

NOTE: You can get only LDAP based attributes in a scope.

Perform the following steps to define scopes and permissions:

1. In the Administration Console, click **Devices** > **Identity Server** > **Edit** > **OAuth & OpenID Connect** > **Resource Server**.
2. Select the resource server name for which you want to define a new scope.
3. Click **New**.
4. Specify the following details:

Field	Description
Name	Specify a name for the scope.
Description	Specify a description for the scope. The consent page shows this description.
Require user permission	Select this option if this scope requires the user's permission before providing access to the protected resources.
Contains user's claims	Select this option to include user's claims in this scope.
Allow modification in consent	Select this option to allow modification in consent. When selected, the resource owner can choose not to share this scope with the client application.

5 Click **Next**.

6 Based on whether you have selected **Require user permission**, you require to perform any one of the following actions:

- **Select an attribute set:** If you have selected the **Require user permission** option, the Identity Server will fetch the required information from the user endpoint. You must specify the LDAP attributes. You can select an attribute set from the list or create a new attribute set. For more information about how to create an attribute set, see "Configuring Attribute Sets" on page 61.
- **Create Claims/Permissions:** If you have not selected **Require user permission**, create claims for the scope.
- Click **New**, specify a name for the claim, and click **OK**.

7 Click **Finish**.

Restricting the Number of Requests

You can restrict the number of users accessing a service by updating the `tomcat.conf` file.

Open `/opt/novell/nam/idp/conf/tomcat.conf`. Add the following parameter:

```
JAVA_OPTS="${JAVA_OPTS} -Dcom.novell.oauth.threshold.maxrequestsallowed=<number of requests>"
```

For example, `JAVA_OPTS="${JAVA_OPTS} -Dcom.novell.oauth.threshold.maxrequestsallowed=10"`. It will not allow more than 10 request per second.

Modifying Scopes of a Resource Server

You can modify the scopes of a registered resource server. Access Manager allows you to delete a resource server or delete the scope of a resource server.

To modify scopes of a resource server, perform the following steps:

1 In the Administration Console, click **Devices** > **Identity Server** > **Edit** > **OAuth & OpenID Connect** > **Resource Server**. This page lists all registered resource servers.

2 Click the *resource server* > *scope* you want to modify.

3 On the Edit Scope page, modify the details as required. For more information about the fields on this page, see "Defining Scopes and Claims for a Resource Server" on page 549.

4 Click **OK**.

Modifying Claims and Attributes

You can modify or delete a defined claim. You can also update the attributes associated with a scope. If you have selected **Require user permission** while creating the scope, the Identity Server fetches the required information from the user endpoint. In this case, you can change the associated LDAP attributes.

If you have not selected **Require user permission**, the scope contains claims instead of LDAP attributes. You can change the name of the claim. For more information about the fields on this page, see "Defining Scopes and Claims for a Resource Server" on page 549.

Managing OAuth Client Applications

A client application that sends API requests to the authorization server must be registered with the authorization server. As part of the registration, specify the client name, redirections (URIs), and any other provider-specific data required by the API. You can register a client application in the Administration Console or in the Identity Server.

Prerequisites for managing client applications include:

- Administration Console: The user must have the NAM_OAUTH2_ADMIN role defined in the OAuth policy.
- Identity Server: The user must have the NAM_OAUTH2_DEVELOPER role defined in the OAuth policy.

An application developer must log into the Identity Server for registering a client application.

The **My Applications** tab lists all the applications that you added. You can view details, modify, and delete applications.

Registering OAuth Client Applications

Perform the following steps to register a client application:

1. In the Administration Console, click **Devices** > **Identity Server** > **Edit** > **OAuth & OpenID Connect** > **Client Applications** > **Register New Clients**.
2. Specify the following details:

Field	Description
Client Name	Specify the name of the client application.
Client Type	Select whether this is a web-based or a desktop client application. For web-based applications specify the client type in this format: https://client.example.org/callback For native/desktop applications, specify the client type in any one of the following formats: https://www.namnetiq.in/ or x-com.netiq.sample://www.namnetiq.in/
Redirect URIs	Specify the URIs that the Identity Server uses to send the authorization code and implicit requests.
Grants Required	Select the grant types required for this client application. Available grant types include: ◆ Authorization Code (default) ◆ Implicit ◆ Resource Owner Credentials ◆ Client Credentials

Field	Description
Token Types	Select the token type that the authorization server will return to this client application. The following are available tokens: ◆ Code ◆ ID Token ◆ Refresh Token ◆ Access Token

3 Click **Consent Screen Configuration**.

Specify the following details:

Field	Description
Client Logo URL	Specify the URL of the logo that you want to include in the consent page.
Privacy Policy URL	Specify the URL of the privacy policy you want to include in the consent page. You can define your own privacy policy.
Terms of Service URL	Specify the URL of the terms of service.
Contacts	Specify email addresses of people related to this client application.

4 Click **Authorized JavaScript origins (CORS)** and add **Domains**. The domains configured here can access restricted resources available on the client application. This is an optional step.

5 Click **Advanced OpenID Connect**. This is an optional step. Specify the following details:

Field	Description
JSON Web Key Set URI	Specify the URI of the JSON file containing the JSON web keys.
ID Token Signed Response Algorithm	Specify the ID Token Signed Response Algorithm.
ID Token Encrypted Response Algorithm	Specify the algorithm used to encrypt the key.
ID Token Encrypted Response Enc	Specify the algorithm used to encrypt the content.

6 Click **Register Client**.

The Identity Server assigns a client ID and a client secret. To see this ID and secret, go to the list of registered client applications on the Client Application page and click the view icon for this client application.

Modifying Registered Client Applications

To modify a registered client application, perform the following steps:

1 In the Administration Console, click **Devices** > **Identity Server** > **Edit** > **OAuth & OpenID Connect** > **Client Applications**. The page lists all registered client applications along with the following details:

Field	Description
Client Application	Name of the registered application
Application Type	Type of the application: Web or Native/Desktop
Created By	User name of the person who has registered the client application.
Actions	List of icons associated with actions that you can perform on an application. You can perform the following actions: ◆ View details of a registered client application ◆ Delete a registered client application ◆ Modify details of a registered client application.

2 Click the edit icon under **Actions**. The Client Configuration page opens. Modify the details as required. For more information about fields, see "Registering OAuth Client Applications" on page 551.

3 Click **Modify Client**.

Configuring the Access Gateway for OAuth

You can configure the Access Gateway to validate OAuth tokens on behalf of the resource server. If the token is not valid then the Access Gateway returns the unauthorized 401 error to the client application.

Configuring the Access Gateway for OAuth consists of the following three steps:

1. Enabling OAuth in the Access Gateway
2. Configuring an Authorization Policy based on OAuth Scopes
3. Configuring an Identity Injection Policy for OAuth Claims or Configuring an Identity Injection Policy for User Passwords

Enabling OAuth in the Access Gateway

If you want the Access Gateway to validate a token before granting access to a protected resource, you must enable OAuth for that protected resource.

Perform the following steps to enable OAuth in the Access Gateway:

1 In the Administration Console, click **Devices** > **Access Gateway** > **Edit** > [*Reverse Proxy name*] > [*Proxy Service name*].

2 Select the **Protected Resources** tab.

3 Click the protected resource for which you want to enable OAuth.

4 Select **Validate OAuth Token**.

5 Click **OK**.

Configuring an Authorization Policy based on OAuth Scopes

You must configure an authorization policy and then assign it to the protected resource. Access Gateway makes decisions based on the rules defined in the authorization policy after validating the OAuth tokens.

Resources protected by OAuth tokens do not execute any authentication procedure. Hence, evaluation of policies associated with OAuth protected resources cannot fetch any user attributes outside the OAuth scope. All the user attributes needed for the protected resource must be part of the OAuth scope. Ensure that the proxy services protected by OAuth are not associated with any policies that refer to authentication contract, profiles, LDAP attribute, LDAP OU, roles, or RISK score. Any policy, which requests for data other than the scope of OAuth token fails.

Perform the following steps to configure an Authorization policy for scopes:

1. In the Administration Console, click **Devices** > **Access Gateway** > **Edit** > [*Reverse Proxy name*] > [*Proxy Service name*].
2. Select the **Protected Resources** tab.
3. Click the protected resource for which you want to configure an Authorization policy.
4. Select the **Authorization** tab.
5. Click **Manage Policies** > **New**.
6. Specify a name for the policy and select **Access Gateway: Authorization** for the policy type.
7. Click **OK**.
8. Specify the following details:

Field	Action
Description	(Optional) Describe the purpose of this rule.
Priority	Specify the order in which a rule is applied in the policy, when the policy has multiple rules. The highest priority is 1 and the lowest priority is 10. **NOTE:** If two rules have the same priority, a Deny rule is applied before a Permit rule.
Conditions	Click **New** and then select **OAuth Scopes**. For **Value**, select the scope from the list.
Actions	Select one of the following: ◆ **Permit:** Allows the user to access the resource. ◆ **Deny:** Select one of the following deny actions: ◆ Display Default Deny Page: Displays a generic message, indicating that the user has insufficient rights to access the resource. ◆ Deny Message: Allows you to provide a customized message that you want to display to users after denying their access attempts. ◆ Redirect to URL: Allows you to specify a URL to redirect users after denying access. For example: `http://www.example.com` ◆ **Redirect:** Specify the URL to which you want the users to redirect when they meet the conditions of this policy. ◆ **Re-authenticate with Contract:** Allows you to specify an authentication contract used to authenticate the user.

9 Click **OK** > **OK**.

10 Select the policy you created and click **Apply Changes** > **Close**.

The Authorization page of the protected resource opens.

11 Select the Authorization policy and click **Enable** > **OK**.

Configuring an Identity Injection Policy for OAuth Claims

You must configure an Identity Injection policy if you want to send the claims details to the resource server. Claims can include user attributes or permissions.

Perform the following steps to configure an Identity Injection policy for scopes:

1 In the Administration Console, click **Devices** > **Access Gateway** > **Edit** > [*Reverse Proxy name*] > [*Proxy Service name*].

2 Select the **Protected Resources** tab.

3 Click the protected resource for which you want to configure an Identity Injection policy.

4 Select the **Identity Injection** tab.

5 Click **Manage Policies** > **New**.

6 Specify a name for the policy, and then select **Access Gateway: Identity Injection** for the type of policy.

7 Click **OK**.

8 Specify the following details:

Field	Action
Description	Specify the purpose of this policy.
Priority	Specify the sequence in which you want to apply the rule in the policy, if the policy has multiple rules. The highest priority is 1 and the lowest priority is 10.
Action	Click **New**, then select one of the following: ◆ **Inject into Authentication Header**: Inserts the user name and password into the header. Select **OAuth Claims** under **user name** and then select a claim. ◆ **Inject into Custom Header**: Inserts custom names into the custom header. Select **OAuth Claims** under **Value** and then select a claim. ◆ **Inject into Custom Header with Tags**: Inserts custom tags with name/value content into the custom header. Select **OAuth Claims** under **Tag Value** and then select a claim. ◆ **Inject into Query String**: Inserts a query string into the URL for the page. Select **OAuth Claims** under **Tag Value** and then select a claim. ◆ **Inject Kerberos Ticket**: Inserts authentication values from the Kerberos ticket into the custom header. Select **OAuth Claims** under **Value** and then select a claim.

9 Click **OK** > **OK**.

10 Select the policy you created and click **Apply Changes** > **Close**.

11 The Identity Injection page of the protected resource opens.

12 Select the Identity Injection policy and click **Enable** > **OK**.

Configuring an Identity Injection Policy for User Passwords

Ensure that you have enabled the **Allow admin to retrieve passwords** option under **Universal Password Retrieval** in the eDirectory user store for all users, so that the policy can retrieve the password from the user store. Without this configuration, the identity injection policy for user password will not work.

For more information about how to enable the password retrieval in eDirectory, see Universal Password Configuration Options in the NetIQ Password Management Administration Guide.

NOTE: The password retrieval works only with eDirectory.

Perform the following steps:

1. In the Administration Console, click **Devices** > **Access Gateway** > **Edit** > [*Reverse Proxy name*] > [*Proxy Service name*].
2. Select the **Protected Resources** tab.
3. Click the protected resource for which you want to configure an Identity Injection policy.
4. Select the **Identity Injection** tab.
5. Click **Manage Policies** > **New**.
6. Specify a name for the policy and select **Access Gateway: Identity Injection** for the policy type.
7. Click **OK**.
8. Configure the policy with the following details:
   - **Action**: Select **Inject into Authentication Header**.
   - **User name**: Select **OAuth Claims** > **Access Token: User**
   - Password: Select **OAuth Claims** > **Password**
9. Click **OK** > **OK**.
10. Select the policy you created and click **Apply Changes** > **Close**.

    The Identity Injection page of the protected resource opens.
11. Select the Identity Injection policy and click **Enable** > **OK**.

Viewing Endpoint Details

In the Administration Console under **Devices** > **Identity Servers** > **Edit** > **OAuth & OpenID Connect** > **EndPoint Summary**, you can view the following endpoints:

- **Authorization EndPoint:** Enables client applications to interact with the resource owner and obtain an authorization grant. It is located on an authorization server.
- **Registration EndPoint:** Enables registering client applications on the authorization server. It is located on the authorization server.
- **Token EndPoint:** Enables client applications to obtain an Access token by providing its authorization grant or Refresh token. It is located on an authorization server.
- **UserInfo EndPoint:** Provides information about the user associated with the Access token in the standard OpenID Connect format.
- **OpenID Metadata EndPoint:** Provides information about OpenID provider metadata.

NOTE: As per OAuth specifications, endpoints must not accept any non-HTTPS request. However, Access Manager supports non-HTTPS requests also. This is required to enable OAuth in scenarios when Access Manager is deployed behind a third-party SSL accelerator.

OAuth and OpenID Connect Audit Events

Access Manager provides the following OAuth audit events:

- OAuth & OpenID Token Issued
- OAuth & OpenID Token Issue Failed
- OAuth Consent Provided
- OAuth Consent Revoked
- OAuth Client Applications
- OAuth & OpenID Token Validation Success
- OAuth & OpenID Token Validation Failed

For more information, see "Enabling Identity Server Audit Events" on page 837.

Enabling Logging for OAuth and OpenID Connect

To enable logging for OAuth and OpenID Connect events, perform the following steps:

1. In the Administration Console, click **Devices** > **Identity Servers** > **Edit** > **Logging**.
2. Select **Enabled** under **File Logging**.
3. In the **Component File Logger Levels** section, specify any one of the following options for OAuth and OpenID Connect:
   - **Off:** Turns off component file logging
   - **Severe:** Logs serious failures that can stop system processing
   - **Warning:** Logs potential failures that have minimal impact on execution.
   - **Info:** Logs informational events.
   - **Verbose:** Logs static configuration information

     The system logs any configuration errors under one of the primary three levels: Severe, Warning, and Info.
   - **Debug:** Logs events for all of the preceding levels (Severe, Warning, Info, and Verbose)
4. Click **OK**.

Managing Clients Applications by Using REST API

You can programmatically register, view, modify, and delete a client application in Access Manager.

Before performing any of these actions, you must have your role defined as `NAM_OAUTH2_DEVELOPER` or `NAM_OAUTH2_ADMIN` in the OAuth policy.

You can register an application in any of the following two ways:

- Using username and password
- Using Access token: To register a client application by using an Access token, you must have your role defined as `NAM_OAUTH2_DEVELOPER` in the OAuth policy.

Use the Resource Owner flow to get an Access token. The endpoint for Resource Owner flow is https://<*Identity Server URL: Port Number*>/nidp/oauth/nam/token. This endpoint requires the followings parameters to provide an Access token:

Parameter	Value
grant_type	password
username	Application developer's user name
password	Application developer's password
scope	◆ urn:netiq.com:nam:scope:oauth:registration:full (This scope allows you to register, view, modify, and delete client applications.) ◆ urn:netiq.com:nam:scope:oauth:registration:read (This scope provides read-only access)
token endpoint	Identity Server URL: Port Number>/ nidp/oauth/nam/token

This section discusses the following topics:

- "Registering a Client Application by Using REST API" on page 558
- "Modifying a Client Application by Using REST API" on page 560
- "Viewing a Client Application by Using REST API" on page 560
- "Deleting a Client Application by Using REST API" on page 560

Registering a Client Application by Using REST API

To register a client application, the HTTP method value must be POST.

The Identity Server uses the following endpoint for registering a client application:

```
https://<Identity Server URL: Port Number>/nidp/oauth/nam/clients
```

The endpoint requires the following parameters:

Table 5-18 *OAuth Parameters for Client Registration*

Parameter	Value	Required/Optional
client_id	Name of the client application.	Required
application_type	web or native	Optional
response_types	Redirection URI values used by the client application.	Required

Parameter	Value	Required/Optional
grant_types	The following are the supported grant types: ◆ authorization_code ◆ implicit ◆ refresh_token ◆ resource_owner_credentials ◆ client_credentials If you do not specify a grant type, the default grant type is used. The default value is authorization_code.	Optional
response_types	The following are the supported response type: ◆ code ◆ id_token ◆ access_token	Optional
logo_uri	Specify the URL of the logo that you want to include in the consent page. Example: https://client.example.org/logo.png	Optional
policy_uri	URL of the Relying Party Client's privacy policy. Example: https://client.example.org/privacypolicy	Optional
tos_uri	URL of the Relying Party's terms of service Example: https://client.example.org/terms	Optional
contacts	Email addresses of people related to this client application.	Optional
jwks_uri	Specify the URI of the JSON file containing the json web keys. This key set contains signing keys that the relying party uses to validate signatures from the OpenID provider. Example: https://client.example.org/my_public_keys.jwks	Optional
id_token_signed_response_alg	Specify the ID Token Signed Response Algorithm. This algorithm is required for signing the ID token issued to the client.	Optional
id_token_encrypted_response_alg	Specify the algorithm used to encrypt the key.	Optional
id_token_encrypted_response_enc	Specify the algorithm used to encrypt the content.	Optional

Modifying a Client Application by Using REST API

Perform the following steps:

1. Retrieve the client details from the `https://<Identity Server URL: Port Number>/nidp/oauth/nam/clients/<client ID>` endpoint. In the request for retrieving client details, use `GET` as the HTTP method value.
2. Send the update request. In the update request, use `POST` as the HTTP method value.

   The Identity Server uses the following endpoint for modifying a registered client application:

   `https://<Identity Server URL: Port Number>/nidp/oauth/nam/clients/`

   For the list of parameters this endpoint requires for a client application modification, see Table 5-18 on page 558.

NOTE: For updating a client application, you must send the complete xml with all parameters during the update request. If you do not include a parameter in the update xml, the server will not initialize this parameter. For example, if you want to update the response_types parameter, send the updated value for this parameter and existing values for other parameters in the request.

Viewing a Client Application by Using REST API

To view a client application, use GET as the HTTP method value.

You can view a registered client application by using the following two endpoints:

- **https://<Identity Server URL: Port Number>/nidp/oauth/nam/clients/:** To view all clients applications registered by a developer
- **https://<Identity Server URL: Port Number>/nidp/oauth/nam/clients/<client ID>:** To view a specific client application registered by a developer

Deleting a Client Application by Using REST API

To delete a client application, the HTTP method value must be DELETE.

The Identity Server uses the following endpoint for deleting a registered client application:

`https://<Identity Server URL: Port Number>/nidp/oauth/nam/clients/<client ID>`

Managing OAuth 2.0 Resource Server and Scope by Using REST API

You can programmatically register, view, modify, and delete a resource server and scopes in Access Manager.

- “Registering a Resource Server by Using REST API” on page 561
- “Deleting a Resource Server by Using REST API” on page 561
- “Viewing Registered Resource Servers” on page 561
- “Creating a Scope by Using REST API” on page 561
- “Modifying a Scope by Using REST API” on page 562
- “Deleting a Scope by Using REST API” on page 562
- “Viewing Configured Scopes by Using REST API” on page 562

Registering a Resource Server by Using REST API

To register a resource server by using REST API, you must have your role defined as `NAM_OAUTH2_ADMIN` in the OAuth policy. Send an `HTTPS POST` request with the appropriate URI parameters to resource server endpoint URI. The request includes the following information:Scope: `urn:netiq.com:nam:scope:oauth:registration:full`

Resource Server Endpoint: `https://<Identity Server URL: Port Number>/nidp/oauth/nam/resourceservers`

HTTP Method: `POST`

URI Parameter: Include the following parameter:

Parameter	Required	Description
name	yes	Name of the resource server

Deleting a Resource Server by Using REST API

To delete a resource server by using REST API, you must have your role defined as `NAM_OAUTH2_ADMIN` in the OAuth policy. The delete request includes the following details:

Scope: `urn:netiq.com:nam:scope:oauth:registration:full`

HTTP Method: `DELETE`

Resource Server Endpoint: `https://<Identity Server URL: Port Number>/nidp/oauth/nam/resourceservers/<resourceServerName>`

Viewing Registered Resource Servers

To view all registered resource servers by using REST API, you must have your role defined as `NAM_OAUTH2_ADMIN` or `NAM_OAUTH2_DEVELOPER` in the OAuth policy. The request includes the following details:

Scope: `urn:netiq.com:nam:scope:oauth:registration:full`

HTTP Method: `GET`

Resource Server Endpoint: `https://<Identity Server URL: Port Number>/nidp/oauth/nam/resourceservers`

Creating a Scope by Using REST API

To create a scope by using REST API, you must have your role defined as `NAM_OAUTH2_ADMIN` in the OAuth policy. The request includes the following details:

Scope: `urn:netiq.com:nam:scope:oauth:registration:full`

Resource Server Endpoint: `https://<Identity Server URL: Port Number>/nidp/oauth/nam/resourceservers/<resourceServerName>/scopes`

HTTP Method: `POST`

Request URI Parameters:

Parameter	Required	Description
scope	Yes	Name of the scope
scope_description	Yes	Description of the scope. The consent page displays this description while obtaining authorization from the user.
claims	Either claims or attribute_set is required.	List of claims
attribute_set		Attribute name and attribute dn. Sample: { "name" : "jpeg_photo", "dn" : "cn=jpeg_photo,o=novell"}
userPermissionRequired	No	Boolean value. Default value is true.
adminApprovalRequired	Yes	Boolean value. Default value is true, set it to false always.
isGroupOfUserAttributes	No	Boolean value. Default value is false.
allowModifyInConsent	No	Boolean value. Default value is false.

Modifying a Scope by Using REST API

To modify a scope by using REST API, you must have your role defined as `NAM_OAUTH2_ADMIN` in the OAuth policy. The request includes the following details:

Scope: `urn:netiq.com:nam:scope:oauth:registration:full`

Resource Server Endpoint: `https://<Identity Server URL: Port Number>/nidp/oauth/nam/resourceservers/<resourceServerName>/scopes/<scopename>`

HTTP Method: `POST`

Send only those parameters which you want to modify.

Deleting a Scope by Using REST API

To delete a scope by using REST API, you must have your role defined as `NAM_OAUTH2_ADMIN` in the OAuth policy. The request includes the following details:

Scope: `urn:netiq.com:nam:scope:oauth:registration:full`

Resource Server Endpoint: `https://<Identity Server URL: Port Number>/nidp/oauth/nam/resourceservers/<resourceServerName>/scopes/<scopename>`

HTTP Method: `DELETE`

Viewing Configured Scopes by Using REST API

You can view details of all scopes together or a specific scope. To view a scope by using REST API, you must have your role defined as `NAM_OAUTH2_DEVELOPER` or `NAM_OAUTH2_ADMIN` in the OAuth policy.

Viewing Details of a Specific Scope

The request includes the following details:

Scope: `urn:netiq.com:nam:scope:oauth:registration:full`

Resource Server Endpoint: `https://<Identity Server URL: Port Number>/nidp/oauth/nam/resourceservers/<resourceServerName>/scopes/<scopename>`

HTTP Method: `GET`

Viewing Details of All Configured Scopes

The request includes the following details:

Scope Required: `urn:netiq.com:nam:scope:oauth:registration:full`

Resource Server Endpoint: `https://<Identity Server URL: Port Number>/nidp/oauth/nam/resourceservers/<resourceServerName>/scopes`

HTTP Method: `GET`

End User Operations

End users can perform the following actions:

- “User Authorization” on page 563
- “Revoking Authorizations” on page 563

User Authorization

When end users access a client application, they are required to give consent for the application to access their email, basic profile, and any other information. An administrator configures a list of allowed scopes. The Consent page shows only these scopes. For example, if an administrator configures email and basic profile, a user can see only these two scopes in the consent page.

Select the scope that you want the application to access and click **Accept**.

NOTE: Email is a mandatory scope configured by the administrator and all client applications can access this scope by default. You cannot deselect this scope on the consent page.

The client application must remember the scopes a user has provided earlier in the consent. For the next request, the client application must ask for the scopes that the user did not provide in the earlier request. For example, if a client application asks for five scopes and the user provides only three scopes, the client application must remember user's choice and must not ask for the five scopes again unless it is an absolute requirement.

Revoking Authorizations

You can view all authorized client applications with their scopes and claims under the **Authorized Applications** tab. An end user can revoke the consent given earlier to client applications by using the Revoke Consent page.

Log into the Identity Server and go to **Applications** > **Authorized Applications**. You can view details of client applications and revoke the client applications’ access if required.

Configuring the Demo OAuth Application

This application demonstrates how to protect an OAuth enabled application by using Access Manager. This application contains a RESTful web service and a client application that uses this RESTful web service.

The RESTful web service allows you to perform TODO tasks for a hypothetical application. This web service exposes an API to add, modify, and delete tasks on behalf of a user. The client application provides a web interface that uses REST APIs to manage these tasks. The REST service protects REST APIs with OAuth Access tokens issued by a trusted Access Manager OAuth provider.

This demo configuration provides a way to test OpenID connect endpoints such as metadata, userinfo, and tokeninfo endpoints.

Configuring the demo OAuth application includes the following steps:

- "Prerequisite" on page 564
- "Registering the Demo RESTful Service in the Administration Console" on page 566
- "Registering the Demo Client Application" on page 566
- "Running REST Services" on page 567
- "Running the Client Application" on page 568
- "Accessing the Client Application through a Web Browser" on page 568

Prerequisite

Ensure that you meet the following prerequisites before running the demo application:

- Install a Linux operating system such OpenSuSE or SuSE Linux Enterprise System or a Windows operating system.
- Set up Java SE Runtime Environment (http://www.oracle.com/technetwork/java/javase/downloads/index.html) 7 or 8.
- Install a web browser. The recommended browser is Google Chrome.
- Install Access Manager 4.1 and configure basic settings. See NetIQ Access Manager Appliance 4.2 Installation and Upgrade Guide and Chapter 3, "Setting Up a Basic Access Manager Appliance Configuration," on page 49.
- Ensure that you have performed the activities listed in "Access Manager Configuration Checklist" on page 565.
- Download the demo application: OAuth 2.0 Demo Application (https://www.netiq.com/documentation/access-manager-41/resources/OAuth_Demo_App.zip).
- Edit host entries for Access Manager and the demo application. See "Editing Host Entries" on page 564.

Editing Host Entries

If your client and server machines are not in the DNS system, you need to edit the host entries.

Adding Access Manager's Host Entries

If the domain name is not configured for Access Manager, you need to add a host entry to your desktop system.

Perform the following steps:

1. In Access Manager Administration Console, go to **Devices** > **Identity Server** > *cluster name* > **Base URL**.
2. Add this Base URL value to the host files.

   `/etc/hosts`
3. Open the host file and add the host entries.

   For example, `10.0.0.0 namtest.mycompany.com`

Adding the Demo Application's Host Entries

If you access the demo application by using a web browser, you must access it through a DNS name. This is the IP address where your application is hosted. For example, if you run the demo application on local host (127.0.0.1.), then the host entry is:

For example, `127.0.0.1 mydemoclient.mycompany.com`

Access Manager Configuration Checklist

Ensure that you have performed the following actions before running the demo application:

☒ Install the Access Manager Administration Console. See "Installing Access Manager Appliance" in the *NetIQ Access Manager Appliance 4.2 Installation and Upgrade Guide*.

☒ Install the Access Manager Identity Server and import it into the Administration Console. See "Installing Access Manager Appliance" in the *NetIQ Access Manager Appliance 4.2 Installation and Upgrade Guide*.

☒ Create an Identity Server cluster configuration and add the imported Identity Server to that cluster. See "Identity Servers Cluster" on page 52.

☒ Enable OAuth & OpenID Connect in **Administration Console** > **Devices** > **Identity Server** > *cluster* > **General**.

☒ Extend the user store to host a new attribute on User Object named `nidsOAuthGrant`. Scopes use this attribute to store the authorization grants provided by a user. If you use the embedded user store of the Administration Console for authenticating users at the Identity Server, then perform the following steps mentioned in "Extending a User Store for OAuth 2.0 Authorization Grant Information" on page 546.

NOTE: **User Store** under **Identity Server** > *cluster name* > **Local** must have the IP address of the Administration Console.

☒ Create a new certificate with key size 2048 and SHA algorithm set to SHA256. In the Administration Console, go to **Security** > **Certificates** > **New** and specify the name as `oauth2048`. You will need this certificate while accessing OpenID Connect endpoints. In Access Manager, OpenID Connect uses certificate configured in **Identity Server** > *cluster* > **OAuth2 & OpenID Connect** > **Global Settings**. OpenID Connect uses the reverse proxy certificate in Access Manager Appliance.

☒ Perform the following actions under **Identity Server** > *cluster* > **OAuth2 & OpenID Connect** > **Global Settings**:

- Set **Authorization Grant LDAP Attribute** to **nidsOAuthGrant** or the name you specified when you extended the user store.
- Select all **Grant Types** and **Token Types**.

- Click **Support Signing** and choose the certificate you have created for this demo configuration.
- Specify the certificate's algorithm.

Registering the Demo RESTful Service in the Administration Console

The demo application contains a simple RESTful web service. This RESTful web service exposes a REST API to add, modify, and delete tasks. A client application can post a request to create, modify, or delete a task by using a REST API. OAuth 2.0 protects this communication. Therefore, each request must contain an OAuth 2.0 Access token with necessary scope `list_todo`, `create_todo`, `delete_todo` to list tasks, add tasks, and delete tasks respectively. Define these scopes in the Administration Console. Later, the client application can request any of these scopes.

To create the scopes, perform the following steps:

1. In the Access Manager Administration Console, click **Devices** > **Identity Server** > **Edit** > **OAuth & OpenID Connect** > **Resource Server** > **New**.
2. Specify `Todo Service` and click **OK**.
3. Click newly created services (`Todo Service`) > **Scopes**.
4. Click **New** and specify the following details:
   - **Name**: create_todo
   - **Description**: Create Task Items
   - **Require user permission**: Select this option
5. Repeat step 4 and add the following scopes:
   - **Name**: `list_todo`, **Description**: `Access your task list`
   - **Name**: `delete_todo`, **Description**: `Delete your task list`

   In this step, you can select **Allow modification in consent**.
6. Click **OK** > **OK** > **Update All**.

Registering the Demo Client Application

The client application needs to communicate with the Identity Server through the OAuth 2.0 protocol. As per this protocol's specification, the Identity Server must uniquely identify each client application. You must register client application in the Identity Manager.

To register a client application in the Identity Server, the user must have the `NAM_OAUTH2_DEVELOPER` role defined in the OAuth policy. Hence, an administrator needs to create the role in the Administration Console and assign it to the user.

Creating an OAuth 2.0 Developer Role for Registering a Client Application in the Identity Server

Perform the following steps:

1. In the Administration Console, click **Devices** > **Identity Servers** > **cluster** > **General** > **Roles**.
2. Click **Manage Policies** > **New**. Specify the following details:
   - **Name**: `oauth_developers`
   - **Type**: `Identity Server: Roles`
3. Click **OK**.

4 Specify the following details under **Condition Group**:

- **New**: **LDAP Attribute**: **LDAP Attribute**: **cn**
- **Comparison**: **String**: **Equals**
- **Value**: **Data Entry Field**: `admin` or provide your own condition here.

5 Specify the following detail under **Actions**:

- **Activate Role**: `NAM_OAUTH2_DEVELOPER`

6 Click **OK** > **OK** > **Apply Changes**.

7 Select the `oauth_developers` and click **Close**.

8 Select the `oauth_developers` policy again and click **Enable** > **OK** > **Update All**.

Registering a Client Application in the Identity Server

Perform the following steps:

1 Determine the Identity Server's base URL. Go to **Administration Console** > **Identity Server** > *cluster* > **General** > **Base URL**.

2 Launch this base URL (https://<base_url>/nidp/) in a web browser.

3 Log into as an administrator.

4 Go to **Applications** > **My Applications** > **Register New Clients** > **Client Configuration**.

5 Specify the following details:

- **Client Name**: `NetIQ Demo Application`
- **Client Type**: **Web**
- **Redirect Uri**: Specify the following URLs with the host name you specified in "Editing Host Entries" on page 564 (mydemoclient.mycompany.com) and the last one with host name of Access Manager installation (namtest.mycompany.com).

  Add each URL in a separate text box.

  `https://mydemoclient.mycompany.com:9443/_oauth-callback`

  `https://mydemoclient.mycompany.com:9443/_oauth-callback2`

  `https://mydemoclient.mycompany.com:9443/ag/callback`

  `https://mydemoclient.mycompany.com:9443/callback`

  `https://mydemoclient.mycompany.com:9443/oidc/_oauth-callback`

  `https://mydemoclient.mycompany.com:9443/oidc/_oauth-callback2`

  `https://namtest.mycompany.com/nidp/netiq/nam/oauth/nam-oauth-callback.html`

6 Select all grants required and token types options.

7 Click **Register Client**.

8 Click `NetIQ Demo Application` (newly registered client application) and note the values of **Client ID** and **Client Secret**.

9 Log out of the Identity Server.

Running REST Services

Now, the demo application is ready to run.

1 Launch the resource server (Task Service).

2 Open a command editor. (Windows: cmd, Linux: any terminal)

3 Verify that the JAVA class path is set correctly:

   Windows: set PATH=c:\Program Files\Jdk\bin;%PATH%

   Linux: export PATH=/usr/lib64/jvm/jdk1.8/bin:$PATH

   Replace the path of JDK wherever it is installed.

4 Locate the downloaded demo application file.

5 Extract `todo-service-0.1-SNAPSHOT.zip`.

6 Go to `todo-service-0.1-SNAPSHOT`

7 Open the Identity Server Base URL (https://namtest.mycompany.com/nidp/bin/todo-service -Dhttp.port=9001)

   Replace the host name with the Identity Server URL. This will run the todo REST service in the port 9001.

Running the Client Application

Perform the following steps:

1 Launch a command editor. (Windows: cmd, Linux: any terminal)

2 Ensure that the Java path is correct. Verify this by running `java`. If not, set the path to JAVA_HOME's bin:

   Windows: set PATH=c:\Program Files\Jdk\bin;%PATH%

   Linux: export PATH=/usr/lib64/jvm/jdk1.8/bin:$PATH

   Replace with the path of JDK wherever it is installed.

3 Locate the downloaded demo application file.

4 Extract `todo-webapp-0.1-SNAPSHOT.zip`.

5 `Go to todo-webapp-0.1-SNAPSHOT`.

6 Open this Administration Console URL= `https://namtest.mycompany.com:8443/nps IDP_BASE_URL=https://namtest.mycompany.com/nidp OAUTH2_CLIENT_ID=uJaQKe5QIC2RZLx5d53lA6sc1-PMOXI_psOrUABLzVQSktHBGoVLF9bXOJElCJ3yft17CbwJmgMHuaz604i55Q OAUTH2_CLIENT_SECRET=ZcX_SxxwmcTezB9nloCxQzHRFo4Yci-2wmbDmyZNMN0CSkhm6UtraPFPFNzB88iEyA5MqluiDq8vomqGUq8RNQ TODO_SERVICE_PORT=9001 TODO_SERVICE_HOST=localhost ./bin/todo-webapp -Dhttp.port=9000 -Dhttps.port=9443`

   This will run the client web application in the port 9443.

Accessing the Client Application through a Web Browser

You can access the application in a browser by using this URL: `https://mydemoclient.mycompany.com:9443/`.

Configuring Authentication Through Federation for Specific Providers

- "Setting Up Google Applications" on page 569
- "Setting Up Office 365 Services" on page 570
- "Integrating Salesforce With Access Manager By Using SAML 2.0" on page 570
- "Integrating Shibboleth Identity Provider With Access Manager" on page 572

Setting Up Google Applications

Google Applications are pre-configured to establish federation with external service providers.

1 In the Administration Console, click **Devices > Identity Servers > Edit > [Protocol]**.

For the protocol, click **SAML 2.0**.

2 Click **New** > **Service Provider**.

NOTE: By default, the **Provider Type > General** is selected.

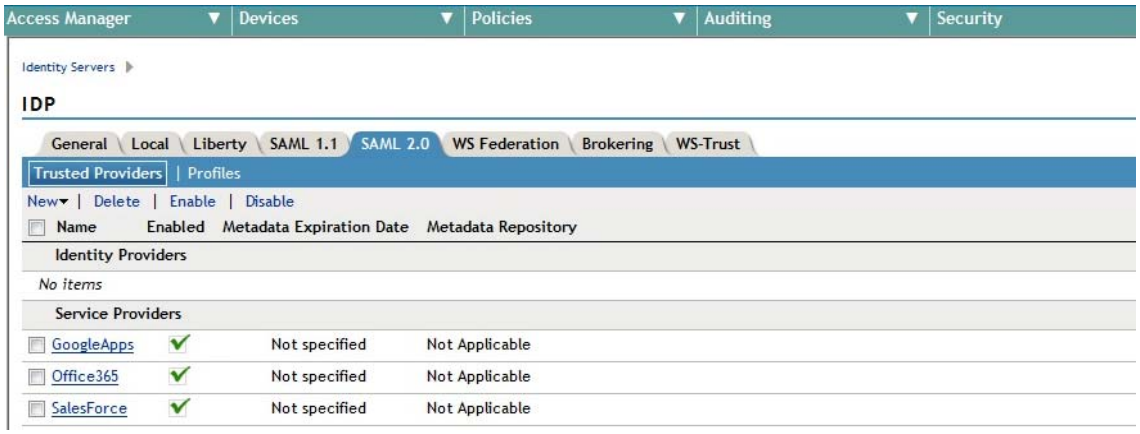
3 Select **Google Application** from the **Provider Type** drop-down list.

By default, the **Metadata Text** source is selected and the **Text** field is pre-filled with the metadata XML. You must edit the Location in the metadata text and replace YOURDOMAIN with the domain name configured in Google Applications.

4 In the **Name** option, specify a name by which you want to refer to the provider and click **Next**.

5 Review the metadata certificates and click **Finish**. For Google Applications, the certificates page displayed is empty because the metadata does not contain information about the certificates. The system displays the trusted provider on the protocol page. For example, if you have specified the **Name** as **GoogleApps**, the page displays the trusted service provider when you click **Finish**.

Figure 5-28 *Trusted Service Provider for Google Application/Office 365/Sales Force*



6 Click **OK**, then update the Identity Server.

The wizard allows you to configure the required options and relies upon the default settings for the other federation options. For information about how to configure the default settings and how to configure the other available options, see Section "Modifying a Trusted Provider" on page 154.

You can configure Access Manager to provide the single sign-on services to Google applications by using Security Assertion Markup Language (SAML) 2.0. For more information, see Integrating Google Apps and Novell Access Manager using SAML 2.0.

Setting Up Office 365 Services

Office 365 is pre-configured to establish federation with external service providers. For more information, see “Setting Up Google Applications” on page 569. In Step 3 on page 569, select **Office 365**. The system displays the trusted provider on the protocol page. For example, if you have specified the **Name** as Office365, the screen displays the trusted service provider **Office365** as in Figure 5-28 on page 569, when you click **Finish**.

Access Manager is compatible with Microsoft Office 365 and provides single sign-on access to Office 365 services.

For more information, see Chapter , “Configuring Single Sign-On for Office 365 Services,” on page 575.

Integrating Salesforce With Access Manager By Using SAML 2.0

Salesforce.com is pre-configured to establish federation with external service providers.

Integrating Salesforce With Access Manager By Using SAML 2.0 for Identity Provider Initiated Login

To integrate Salesforce for idpsend, follow the procedure in “Setting Up Google Applications” on page 569. In Step 3 on page 569, select **Salesforce**. The system displays the trusted provider on the protocol page. For example, if you have specified the **Name** as SalesForce, the screen displays the trusted service provider as in Figure 5-28 on page 569, when you click **Finish**.

Access Manager allows your users to use their existing LDAP credentials for single sign-on access to salesforce.com as well as any Web applications protected by Access Manager.

For information using SAML 2.0 for Identity Provider initiated login, follow the procedure below.

1. Create domain in Salesforce.

   To enable IDP-initiated login in Salesforce.com, you must enable and configure the **My Domain** option in Salesforce.com. Defining your own domain provides the basis for an IDP-initiated URL.

   1a Login as administrator. Go to **Administration Setup** > **Domain Management** > **My Domain**.

   1b Specify the subdomain name and check the availability.

   1c Agree to the terms and conditions and click **Register Domain**.

2. If you have already configured your identity provider for Salesforce.com using the wizard, you must update configuration in the identity provider according to the new domain. Perform the following steps.

   2a Download the metadata from Salesforce site for your domain. See Step 3 on page 569. Send and import this metadata into your Identity Server Salesforce configuration. For reimporting metadata in Access Manager Identity Server, see “Viewing and Reimporting a Trusted Provider’s Metadata” on page 158.

   2b Change the Intersite Transfer URL to point to the new domain URL

3. Perform Step 4 on page 571 and Step 5 on page 572.

4. Update the Identity Server.

Integrating Salesforce With Access Manager By Using SAML 2.0 for Service Provider Initiated Login

Service provider configuration options offer you more flexibility and control for example, simultaneously federating with more than one Identity Server. Salesforce.com also supports SP-initiated login along with IDP-initiated login. SP-initiated login lets the user use a simple and intuitive URL to access the target application.

Follow the procedure given below to integrate Salesforce with Access Manager by using SAML 2.0 for service provider initiated login. Assume that the user has a Salesforce account.

1 Create domain in Salesforce.

To enable SP-initiated login in Salesforce.com, you must enable and configure the **My Domain** option in Salesforce.com. Defining your own domain provides the basis for an SP-initiated URL.

1a Login as administrator. Go to **Administration Setup** > **Domain Management** > **My Domain**.

1b Specify the subdomain name and check the availability.

1c Agree to the terms and conditions and click Register Domain.

If you have already configured your identity provider for Salesforce.com using wizard, you must update configuration in the identity provider according to the new domain. Perform the following steps.

NOTE: Configure SSO configuration. Follow the procedure below to enable SAML support in Salesforce.

1. Go to your Salesforce account and login.
2. From the left panel, select **Security Control** > **Single sign setting** > **Saml Single Sign-on Setting** > **New** and fill the form.
3. To enable SAML select **Security Control** > **Single sign setting** > **Saml Single Sign-on Setting** > **Federated Single Sign-On Using SAML** > **Edit** > **Enable Saml**.

---

2 Change the Intersite Transfer URL to point to the new domain URL.

3 Import Salesforce metadata in Access Manager.

As with any other SAML federation you must configure both your Access Manager Identity Server and Salesforce.com Service Provider (SP) to establish a trust. You now have an option to download your metadata from Salesforce.com. To download your specific metadata go to your Salesforce.com instance.

3a Login as administrator. Go to **Administration Setup** > **Security Controls** > **Single Sign-On Settings**.

3b Select **Name** which you have configured above and **Download Metadata**.

3c Reimport this metadata into your service provider configuration in Access Manager assuming that you have created Salesforce using the wizard.

The metadata file you download will include a certificate. For Access Manager to trust or use this certificate, the trusted root certificate chain that minted the certificate must exist in the Access Manager certificate trust stores.

4 Import certificate in Access Manager, for example, Salesforce.com.

4a Open the downloaded metadata `.xml` file with a file editor and search for the certificate in the X509Certificate element (between <ds:X509Certificate> and </ds:X509Certificate>).

4b Copy the information into its own file and give it a `.cer` file extension. Windows will recognize this as a certificate.

4c Double click and open the file.

4d Click **Certification Path** to see the chain of authority for the certificate.

You will need the trusted root certificate for every CA in the chain that you see listed.

4e In the example above, select the **VeriSign Class 3 International Server CA – G3** and click **View Certificate**.

4f Click **Details**.

5 You can now export the CA trusted root certificate.

5a Click **Copy to File…**. This will launch the Windows Certificate Export Wizard.

5b Select **.DER** encoded when prompted. Give the file a name and save.

5c Repeat this process for every CA in the certificate path chain.

5d Use the Access Manager Administration Console to import the resulting CA trusted root certificates into your Access Manager keystores.

Ensure to add Root certificate of Salesforce into your OCSP trust store else, OCSP validation fails and the Identity Server displays an error.

Integrating Shibboleth Identity Provider With Access Manager

You can establish a single sign-on exchange between Access Manager SAML 2 service provider and a Shibboleth SAML 2 identity provider.

For more information, see Integrating Access Manager with Shibboleth's Identity Provider Server.

Integrating Amazon Web Services with Access Manager

NetIQ Access Manager now enables you to federate with the Amazon Web Services (AWS) with the help of a wizard. The wizard allows you to configure the required options and relies upon the default settings for the other federation options.

Integrating AWS with Access Manager includes the following steps:

- "Enabling Web Single Sign-On in the AWS Console" on page 572
- "Configuring AWS as a Service Provider in Access Manager" on page 573

Enabling Web Single Sign-On in the AWS Console

Before you integrate AWS in Access Manager, you must enable web single sign-on (sso) in the AWS console. To enable web sso, perform the following steps:

1 Download the Access Manager SAML 2.0 metadata by accessing `https://<www.idp.com:8443>/nidp/saml2/metadata`. Save into local file and rename it as `nam-saml2-metadata.xml`.

2 Open the AWS console by accessing: https://console.aws.amazon.com/console/home.

3 Log in to AWS.

4 Click **Security & Identity > Identity & Access Management**.

5 Click **Identity Providers**.

6 Click **Create Provider**.

6a **Provider Type**: Select **SAML**.

6b **Provider Name**: Specify a name. For example, NAM-IDP.

6c **Metadata Document**: Select the file that you saved in Step 1 on page 572.

7 Verify the provider information and click **Create**.

8 On the dashboard, click **Roles**.

9 Click **Create New Role**.

10 Specify a role name.

11 Click **Next**.

12 Select **Role for Identity Provider Access** > **Grant Web Single Sign-On (WebSSO) access to *[SAML providers]***.

13 Click **Next Step**.

14 On the **Attach Policy** page, select the desired policies. Click **Next Step**.

15 Review the role information. Make a note of the Role ARN and Trusted Entries.

16 Click **Create Role**.

Configuring AWS as a Service Provider in Access Manager

1 In the Administration Console, click **Devices > Identity Servers > Edit > SAML 2.0**.

2 Click **New > Service Provider**.

3 Specify the following details:

Provider Type: Select **Amazon Web Services**.

By default, the **Metadata Text** source is selected and the **Text** field is pre-filled with the metadata XML.

Name: Specify a name for the provider and click **Next**.

Role ARN: Specify the role ARN. For example, specify `arn:aws:iam:625143326143:role/MyAdmin`.

Trusted SAML Provider ARN: Specify the trusted SAML provider ARN. For example, specify `arn:aws:iam:625143326143:saml-provider/idp1`.

To fetch ARN values, see "Enabling Web Single Sign-On in the AWS Console" on page 572.

NOTE: The Role ARN and Trusted SAML Provider ARN parameters are used to create the attribute mapping. If you have configured multiple roles in AWS, you can add any Role ARN while creating a service provider. To modify the attribute set, see "Re-Mapping Attribute Sets" on page 573.

---

4 Review the metadata certificates and click **Finish**.

5 Click **OK**, then update the Identity Server.

Re-Mapping Attribute Sets

By default, the AWS wizard creates an attribute set with the name `AmazonWebServices`. This attribute set has the following mappings:

1 **Constant Value**: It is created using the Role ARN and trusted SAML provider. It is mapped to Role.

For example: if Role ARN is `arn:aws:iam::638116851885:role/NewRole` and the Trusted SAML Provider ARN is `arn:aws:iam::638116851885:saml-provider/NAM-IDP`, then, the constant value is `arn:aws:iam::638116851885:role/NewRole,arn:aws:iam::638116851885:saml-provider/NAM-IDP`. This is mapped to the Role.

NOTE: When multiple roles are configured in AWS, create a virtual attribute to change the Role ARN dynamically depending on the user. After creating the virtual attribute, create the corresponding attribute mapping. For more information, see use case 3 in "Sample JavaScripts with Examples" on page 79.

2 **LDAP Attribute**: It is the `givenName` mapped to the Remote Attribute `RoleSessionName`. You can also map any other attribute instead of the givenName.

If you want to use any other LDAP attribute to be mapped for `RoleSessionName`, perform the following steps:

1 In the Administration Console, click **Devices > Identity Server > Shared Settings > Attribute Sets > AmazonWebServices > Mapping**.

In the attribute list, select the existing LDAP attribute set.

2 Click **Delete**.

3 Click **Apply > OK**.

4 Click **New**.

5 In **Add Attribute Mapping**, specify the following details:

5a **Local attribute**: Select a local attribute from the available list.

5b **Remote Attribute**: Specify RoleSessionName.

5c **Remote nameSpace**: Specify http://aws.amazon.com/SAML/Attributes/

6 Click **OK > Finish**.

7 In the Administration Console, click **Devices > Identity Servers > Edit > SAML 2.0**.

8 Select **AWS** and click **Attributes**.

9 Select the new attribute set from **Available** and move it to **Send with authentication**.

10 Click **OK**, then update the Identity Server.

Re-Importing The Metadata

The AWS metadata has a validity associated with it. You need to re-import the metadata before the license expires. To re-import the metadata, perform the following steps:

1 In the Administration Console, click **Devices > Identity Servers > Edit > SAML 2.0**.

2 Under Trusted provider, click AWS service provider.

3 In **Metadata**, click **Reimport**.

4 Specify the following:

4a **Provider Type**: Select General.

4b **Source**: Select Metadata text.

4c **Name**: Name for the service provider is displayed by default.

4d **Text**: Fetch the metadata from: `https://signin.aws.amazon.com/static/saml-metadata.xml`. Remove the string content `<KeyDescriptor use="signing"> .... </KeyDescriptor>`. Copy this edited metadata and paste it in **Text**.

5 Click **Next**.

6 Confirm metadata certificates, then click **Finish**.

7 Update the Identity Server.

Configuring Single Sign-On for Office 365 Services

NetIQ Access Manager provides single sign-on access to Office 365 services such as Exchange Server, Sharepoint Online and Lync without using ADFS (Active Directory Federation Services). You can use your existing enterprise credentials to access any of the Office 365 services without having to remember multiple passwords or sign in multiple times to access different services. You can sign in once with an existing password and Access Manager grants you access to all services.

This single sign-on access is achieved by implementing Passive or Active authentication by using WS-Federation, WS-Trust, and SAML 2.0 protocols.

A trust model is set up for Access Manager and Office 365 to communicate with each other. Access Manager, configured as an identity provider, allows Office 365 to trust it for authentication. Office 365 configured as a service provider, consumes authentication assertions from Access Manager.

Access Manager supports single sign-on to the following Office 365 applications:

- SharePoint
- Office 365 Portal
- Outlook Web Access
- Lync 2010
- Lync 2013
- Skype for Business 2015
- Outlook 2013

Topics include:

- “Passive and Active Authentication” on page 575
- “Configuring Active and Passive Authentication By Using WS-Trust and WS-Federation Protocols” on page 576
- “Configuring Federation with Office 365 Services for Multiple Domains” on page 579
- “Configuring an Office 365 Domain That Supports Passive Federation by using SAML 2.0” on page 581
- “Useful Resources” on page 586
- “Troubleshooting Scenarios” on page 586
- “Sample Tokens” on page 589

Passive and Active Authentication

In a Passive authentication scenario, the user signs in through a Web form displayed by the identity provider and the user is requested to log in. In Active authentication scenario, the user is authenticated using thick clients. As the thick client does not support redirection, Office 365 gets the credentials and validates the authentication with Access Manager by communicating directly with it.

Passive authentication is supported by using the WS-Federation protocol and supports sign-in to Office 365 using the Web interface. The clients includes the Office 365 portal, SharePoint Online, Outlook Web Access, and the Office Web Apps. You can achieve passive authentication using either SAML 2.0 or WS-Federation protocol.

Active authentication is supported by using the WS-Trust protocol and supports sign-in to Office 365 using Office client applications. The clients includes Outlook, Lync, Word, Excel, PowerPoint, and OneNote. If you are using Microsoft Exchange, you can use SAML 2.0 but for active authentication, WS-Trust is the recommended protocol.

Configuring Active and Passive Authentication By Using WS-Trust and WS-Federation Protocols

Using the wizard, when you configure an Office 365 domain with WS-Trust protocol it creates the following two domains:

- A domain preconfigured for active authentication using WS-Trust protocol
- A domain preconfigured for passive authentication using WS-Federation protocol.

If your business needs demand using a domain based on SAML 2.0 protocol, you can configure a domain manually using the steps in “Configuring an Office 365 Domain That Supports Passive Federation by using SAML 2.0” on page 581

The following sections cover the details about how to configure a domain by using WS-Trust and WS-Federation protocols:

- “Prerequisite” on page 576
- “Configuring an Office 365 Domain By Using WS-Trust Protocol” on page 576
- “Configuring an Office 365 Domain to Federate with Access Manager” on page 577

Prerequisite

Use the following steps to verify that WS-Trust and WS-Federation protocols are enabled in Access Manager:

1. In the Administration Console, click **Devices** > **Identity Servers** > **Edit**.
2. In the **Enabled Protocols** section, ensure that **WS-Trust** and **WS-Federation** protocols are selected.

Configuring an Office 365 Domain By Using WS-Trust Protocol

When you configure a new Office 365 domain by using the WS-Trust protocol, it creates a domain preconfigured for Active authentication and also creates a WS-Federation Service Provider that is preconfigured for Passive authentication.

1. In the Administration Console, click **Devices > Identity Servers > Edit > WS-Trust > Service Provider Domain**.
2. Click **New > Office 365 Domain** and specify a name to identify the domain. This domain is by default configured with ImmutableID and Attribute Set information and a Service Provider with the same name as the Office 365 domain is automatically created.

   An authentication method `Name/Password - Form-WebService` is created and this is selected for WS-Trust. This method ensures that an email address/password is accepted for authentication.

   Click the domain name to make further modifications.

   For more details, see link “Modifying Service Providers” on page 524
3. Click the **WS-Federation** tab and verify that a new Service Provider with the same name as the Office 365 domain is created. This Service Provider is preconfigured with Attribute Set information and Authentication Response for the Passive authentication.

Configuring an Office 365 Domain to Federate with Access Manager

- "Prerequisite" on page 577
- "Enabling Federation Settings in Office 365 Domain" on page 577
- "Verifying Single Sign-On Access" on page 578

Prerequisite

Ensure that the following requirements are met before configuring an Office 365 domain:

- The Identity Server must be accessible from outside the firewall so that the Office 365 domain can communicate with the Identity Server.
- Sign up for an Office 365 account.
- To single-sign on to any of the Office 365 applications, ensure that you download it from the Office 365 portal.
- Create a federated domain in Office 365 and prove ownership of it. This ensures that you add your company domain into the Office 365 domain. For more information, see Adding and Verifying a Domain for Office 365.
- Ensure that the Windows 7 or Windows 8 workstations do not have the Active Directory Federation Service 2.0 snap-in installed.
- Ensure that the Identity Server Base URL SSL certificate is issued by a well-known external certification authority (CA).
- Install Microsoft Live Sign-in Module to help manage and establish a remote session with the Office 365 account that is created to manage the Office 365 domain. To download, go to Microsoft Downloads Center.
- Install Microsoft Azure Active Directory Module. To download, go to Manage Azure AD using Windows PowerShell.

Enabling Federation Settings in Office 365 Domain

Run the following commands in Powershell by modifying the commands with your domain name as per your setup. The domain name in the example is `namtest.com`.

1. From the Start Menu launch Windows Azure Active Directory Module for Windows PowerShell.
2. Run `$cred=Get-Credential`. Enter your cloud service administrator account credentials.
3. Run `Connect-MsolService –Credential $cred`

   For example, if the name of the domain is `namtest.com` and the Base URL of the Identity Server is `https://namtest.com/nidp/`, execute the following commands at the Powershell prompt:

   NOTE: In this example, the Base URL the port is not specified as it uses the default port 443. If you are using a different port, specify the port with the Base URL.

   For example: `https://namtest.com/nidp/`

   ---

   1. `$dom = "namtest.com"`
   2. `$url = "https://namtest.com/nidp/wsfed/ep"`
   3. `$ecpUrl = "https://namtest.com/nidp/wstrust/sts/active12"`
   4. `$uri = "https://namtest.com/nidp/wsfed/"`
   5. `$logouturl = "https://namtest.com/nidp/jsp/o365wsfedlogout.jsp"`
   6. `$mex = "https://namtest.com/nidp/wstrust/sts/mex"`

7. `$cert = New-Object System.Security.Cryptography.X509Certificates.X509Certificate2("name and path of the certificate")`

   NOTE

   - If the certificate used has a `.crt` extension ensure that you convert it to a `.cer` extension file.
   - While executing this command, ensure that you specify the path to the certificate within the double quotes. For example: "`C:\local\netiq-off365-sign.cer`"

8. `$certData = [system.convert]::tobase64string($cert.rawdata)`

4 Use the following cmdlet to update the settings of the single sign-on domain.

```
Set-MsolDomainAuthentication -FederationBrandName $dom -Authentication
Federated -PassiveLogOnUri $url -SigningCertificate $certData -IssuerUri $uri -
ActiveLogOnUri $ecpUrl -LogOffUri $logouturl -MetadataExchangeUri $mex
```

Verifying Single Sign-On Access

Prerequisite:

- You need at least one user in Office 365 to verify that single sign-on is set up. If you have an existing user, ensure that the Immutable ID matches the GUID of the Access Manager user.

  For instance if your user store is eDirectory and you want to retrieve the GUID of an existing Access Manager user, execute the following command on the eDirectory server terminal:

  ```
  ldapsearch -x -h 127.0.0.1 -p 389 -D cn=admin,o=novell -w novell -b
  cn=anand,o=novell guid
  ```

  Where h is host, p is port, D is bind credential, w is password, b is search scope, and guid is the attribute to search.

  Create an Office 365 user with this GUID as the Immutable ID using the following command in Powershell:

  ```
  new-msolUser -userprincipalName "user1@domain name" -immutableID "GUID of
  user1" - lastname "lastname of user 1" -firstname user1 -DisplayName "user1
  users" -BlockCredential $false -"LicenseAssignment testdomain:ENTERPRISEPACK"
  -usageLocation "two letter country code[example: US,IN,DE,BE,GB etc]" -Password
  "password of the user" - LicenseAssignment validlicense.
  ```

Procedure to verify:

To verify that single sign-on is set up correctly, perform the following procedure in a server that is not added to the domain.

1 Go to Microsoft Online Services (http://login.microsoftonline.com/)

2 Log in with your corporate credentials. (For example : `user1@namnetiq.in`)

  If single sign-on is enabled, the password field is dimmed. You will instead see the following message: `You are now required to sign in at <your company>`.

3 Select the **Sign in at your company** link.

  If you are able to sign in without errors, single sign-on is set up successfully.

Configuring Federation with Office 365 Services for Multiple Domains

You can now federate multiple parent domains with a single Access Manager cluster. This means that if the enterprise has users belonging to multiple domains, a single Access Manager cluster can handle the single sign-on requests for all the users for Office 365 services.

For example: Let us assume you have users spread across two domains: `user1@namtest.com` and `user2@namnetiq.in`. When `user1@namtest.com` and `user2@namnetiq.in` access Office 365 services, the same Access Manager identity provider automatically forms the response with the corresponding Issuer URI and sends it to corresponding domains configured in the Office 365 service.

- “Creating Multiple Domains in Office 365 and Establishing Federation with Access Manager” on page 579
- “Configuring Federation for Multiple Domains that Include Child Domains” on page 580

Creating Multiple Domains in Office 365 and Establishing Federation with Access Manager

1. Ensure that you meet the prerequisites for creating a domain. For more information, see “Prerequisite” on page 576.
2. Create a new Office 365 domain and verify it. For more information see Adding and Verifying a Domain for Office 365. (http://office365support.ca/adding-and-verifying-a-domain-for-the-new-office-365/)

   NOTE: Office 365 does not support creating a child domain if federation configuration for parent domain is already established by using powershell. Ensure that you add all child domains from the Office 365 admin center before establishing federation for the parent domain.

   For more information about establishing federation when there are multiple domains and a child domain, see “Configuring Federation for Multiple Domains that Include Child Domains” on page 580.

   ---

3. According to the example used in section *Enabling Federation Settings in Office 365 Domain,* we have an existing domain named `namtest.com`.

   To create a new domain named `namnetiq.in`, run the following commands in Powershell by modifying the commands with your domain name as per your setup.

   3a Run `$cred=Get-Credential`. Enter your cloud service administrator account credentials.

   3b Run `Connect-MsolService –Credential $cred`

   For example, if the name of the domain is `namnetiq.in` and the Base URL of the Identity Server is `https://namnetiq.in/nidp/`, execute the following commands at the Powershell prompt:

   NOTE

   - In the following example, port is not mentioned as it uses 443. However, if you are using port 8443, specify the port with the Base URL as follows:

For example: `https://namnetiq.in:8443/nidp/`

- When you add additional domains to Office 365 using Powershell commands, the variables `$certdata`, `$url`, `$ecpurl`, `$logouturl`,and `$mex` must contain the details provided for the existing domain. If you configure a new domain, change the values of `$dom` and the `$uri`

---

1. `$dom = "namnetiq.in"`
2. `$url = "https://namtest/nidp/wsfed/ep"`
3. `$ecpUrl = "https://namtest.com/nidp/wstrust/sts/active12"`
4. `$uri = "https://namnetiq.in/nidp/wsfed/"`
5. `$logouturl = "https://namtest.com/nidp/jsp/o365wsfedlogout.jsp"`
6. `$mex = "https://namtest.com/nidp/wstrust/sts/mex"`
7. `$cert = New-Object System.Security.Cryptography.X509Certificates.X509Certificate2("`*`name and path of the certificate`*`")`

   NOTE

   - If the certificate used has a `.crt` extension ensure that you convert it to a `.cer` extension file.
   - While executing this command, ensure that you specify the path to the certificate within the double quotes. For example: “`C:\local\netiq-off365-sign.cer`”

8. `$certData = [system.convert]::tobase64string($cert.rawdata)`

3c Use the following cmdlet to update the settings of the single sign-on domain.

```
Set-MsolDomainAuthentication -FederationBrandName $dom -Authentication
Federated -PassiveLogOnUri $url -SigningCertificate $certData -IssuerUri
$uri -ActiveLogOnUri $ecpUrl -LogOffUri $logouturl -MetadataExchangeUri
$mex
```

To configure any more domains, follow the same steps. Ensure that the Issuer URI includes the UPN of the domain. For example. If you are configuring a domain named `support.in`, the Issuer URI will be `https://support.in/nidp/wsfed/`

4 Go to **Devices > Identity Servers > Edit** > **Options** and ensure that the value for STS OFFICE365 MULTI DOMAIN SUPPORT AUTO is configured as true.

This property enables users to access Office 365 services using the Issuer URI specific to the domain they belong to.

Configuring Federation for Multiple Domains that Include Child Domains

Consider a scenario where you already have users as part of `namtest.com` and `namnetiq.in`. You now have to create a child domain `support.namnetiq.in` under `namnetiq.in`. In this case no federation settings are available in Office 365 for the child domain. The federation setting for the parent domain is used. So, it is important that the Issuer URI is not automatically changed to the User Principal Name of the user. The Issuer URI must be set to the parent domain Issuer URI. For the child domain `support.namnetiq.in`, the Issuer URI will be `https://namnetiq.in/nidp/wsfed/`

1 In the Administration Console, click **Devices > Identity Servers > Edit** > **Options** > **New**.

Property Type	Property Value

STS CHANGE ISSUER	Specify the value in this format: `SPentityID:UPNDomain -> new IssuerID`.
	
	For example,
	`urn:federation:MicrosoftOnline:support.namnetiq.in -> https://namnetiq.in/nidp/wsfed/`
	
	For example,
	`urn:federation:MicrosoftOnline:support.namnetiq.in -> https://namnetiq.in/nidp/wsfed/`
	
	In case of multiple child domains, add each parent domain and child domain separated by comma. For example, if namnetiq.in is the parent domain and support.namnetiq.in and engineering.namnetiq.in are the child domains, specify the following entries:
	
	`urn:federation:MicrosoftOnline:namnetiq.in -> https://namnetiq.in/nidp/wsfed/, urn:federation:MicrosoftOnline:support.namnetiq.in -> https://namnetiq.in/nidp/wsfed/, urn:federation:MicrosoftOnline:engineering.namnetiq.in -> https://namnetiq.com/nidp/wsfed/`

STS OFFICE365 MULTI DOMAIN SUPPORT AUTO	Select false. This ensures that the Issuer URI is formed based on the UPN of the parent domain.

2 Click **OK** > **Apply**.

Configuring an Office 365 Domain That Supports Passive Federation by using SAML 2.0

- “Prerequisite” on page 581
- “Configuring an Office 365 Domain to Federate with Access Manager” on page 581

Prerequisite

Ensure that SAML 2.0 is enabled in Access Manager.

1 In the Administration Console, click **Devices**> **Identity Servers** > **Edit**.

2 In the **Enabled Protocols** section, verify whether SAML 2.0 is selected.

Configuring an Office 365 Domain to Federate with Access Manager

- “Prerequisite” on page 582
- “Setting Up Office 365 Services” on page 582
- “Establishing Trust Between an Identity Provider and a Service Provider” on page 582
- “Configuring Specific Attributes as ImmutableID” on page 583

- “Configuring Desktop Email Client to Access Office 365 Emails” on page 584
- “Verifying Single Sign-On Access” on page 585

Prerequisite

Ensure that the following requirements are met before configuring an Office 365 domain:

- Sign up for an Office 365 account.
- To single-sign on to any of the Office 365 applications, ensure that you download it from the Office 365 portal.
- Create a federated domain in Office 365 and prove ownership of it. By doing this you add your company domain into the Office 365 domain. For more information, see Adding and Verifying a Domain for Office 365.
- Ensure that the Windows 7 or Windows 8 workstations do not have the Active Directory Federation Service 2.0 snap-in installed.
- Ensure that the Identity Server Base URL SSL certificate is issued by a well-known external certification authority (CA).
- Install Microsoft Live Sign-in Module to help manage and establish a remote session with the Office 365 account that is created to manage the Office 365 domain. To download, go to Microsoft Downloads Center.
- Install Microsoft Azure Active Directory Module. To download, go to Manage Azure AD using Windows PowerShell.

Setting Up Office 365 Services

Office 365 is preconfigured to establish federation with an external service providers.

Perform the following steps to create a trusted service provider:

1. In the Administration Console, click **Devices**> **Identity Servers** > **Edit**.
2. Click **SAML 2.0 > New >Service Provider**.
3. Select **Provider Type** as `Office 365`. Ensure that **Source** is selected as the `Metadata Text`.
4. Specify a name for the Office 365 domain. The XML metadata is automatically populated in the **Text** field. Click **Next**.
5. Confirm the certificates and click **Finish** to save the changes.

Establishing Trust Between an Identity Provider and a Service Provider

You can configure Office 365 domains federations by using the Microsoft Online Services Module. You can use the Microsoft Online Services Module to run a series of cmdlets in the Windows PowerShell command-line interface to add or convert domains for single sign-on.

Each Active Directory domain that you want to federate by using Access Manager must either be added as a single sign-on domain or converted to be a single sign-on domain from a standard domain. Adding or converting a domain sets up a trust between Access Manager and Office 365.

Adding a Domain:

To add a domain to Office 365, perform the following steps:

1. Log in to Office 365 as an administrator.
2. On the Administrator page, click **Management** > **Domains** > **Add a domain**.
3. Specify the domain name that you want to add.
4. Click **Next**.

5 Verify the domain name.

   For more information about how to verify a domain, see Verify your domain and change name servers.

6 Select appropriate services.

7 Configure the DNS records on the domain registrar for other services.

NOTE: Do not configure the new domain to the primary domain. Using the `Set-MsolDomainAuthentication` command to set the domain as a federated domain results in an error if the domain is the default domain.

For more information, see Add a domain to Office 365.

Converting a standard domain to a federated domain: To convert a standard domain to a federated domain, perform the following steps:

1 Open the Microsoft Online Services Module from the Start menu.

2 Run `$cred=Get-Credential`. Enter your cloud service administrator account credentials.

3 Run `Connect-MsolService –Credential $cred`.

   This cmdlet connects you to the cloud service. Creating a context that connects you to the cloud service is required before running any of the additional cmdlets installed by the tool.

   For example, if the name of the domain you are converting to a single sign-on domain is *namtest.com*, and the base URL of the Identity Server is *https://namtest.com:8443/nidp*, execute the following commands at the Powershell prompt:

   1. `$dom = "namtest.com"`
   2. `$url = "https://namtest.com:8443/nidp/saml2/sso"`
   3. `$ecpUrl = "https://namtest.com:8443/nidp/saml2/soap"`
   4. `$uri = "https://namtest.com:8443/nidp/saml2/metadata"`
   5. `$logouturl = "https://namtest.com:8443/nidp/jsp/o365Logout.jsp"`
   6. `$cert = New-Object System.Security.Cryptography.X509Certificates.X509Certificate2("name and path of the certificate")`

      NOTE: While executing this command, ensure that you specify the path to the certificate within the double quotes. For example: “`C:\local\netiq-off365-sign.cer`”

      ---

   7. `$certData = [system.convert]::tobase64string($cert.rawdata)`

4 Use the following cmdlet to update the settings of the single sign-on domain:

```
Set-MsolDomainAuthentication -FederationBrandName $dom -Authentication Federated -
PassiveLogOnUri $url -SigningCertificate $certData -IssuerUri $uri -ActiveLogOnUri $ecpUrl -
LogOffUri $logouturl -PreferredAuthenticationProtocol SAMLP
```

NOTE: Ensure that there are no spaces after the hyphen if you are copy pasting the command.

Configuring Specific Attributes as ImmutableID

By default, Office 365 uses GUID as the ImmutableID. If you want to use any other attribute as the ImmutableID of the Office 365 user, configure and then add a property name/value pair.

1 In Administration Console, go to **Identity Server** and select an Identity Server.

2 Select **SAML 2.0** and then select the service provider you created.

3 Select **Options** and click **New**.

4 Add a property name/value pair as follows:

Property Name: SAML2_OFFICE365_NAMEID_ATTRIBUTE_NAME

Property Value: title

The title you specify in the **Property Value** must be base64 encoded and stored in the user store. This value must be used as ImmutableID while creating a user in Office 365.

Configuring Virtual Attributes as ImmutableID

NOTE: Applicable only from Access Manager 4.2.3 onwards.

Virtual attributes can be helpful in many scenarios:

- If the attribute has clear text value, virtual attribute can be used to dynamically convert it into a base64 value
- Retrieve the attribute from an external datasource and use it as the ImmutableID

To use virtual attribute as ImmutableID, perform the following steps:

1 Create a virtual attribute. For information about how to create a virtual attribute, see "Managing a Virtual Attribute" on page 75.

2 Go to **Devices > Identity Servers > Servers > Edit > SAML 2.0 > Service Provider > Options** and add the following property for Office 365:

- **Property Name**: SAML2_OFFICE365_NAMEID_ATTRIBUTE_NAME
- **Property Value**: Specify the name of the virtual attribute

3 Add the virtual attribute created in step 1 to the attribute set created for the SAML 2.0 service provider configuration for Office 365. This ensures that the name specified in SAML2_OFFICE365_NAMEID_ATTRIBUTE_NAME is mapped to the virtual attribute with that value. If the virtual attribute is not added to the attribute set, Access Manager considers the name as a normal LDAP attribute name.

Configuring Desktop Email Client to Access Office 365 Emails

You can configure your desktop email client to access Office 365 emails. The email clients must use a basic authentication and a supported exchange access method such as IMAP, POP, Active Sync, and MAPI.

The following are the list of email clients supported for this configuration:

- Microsoft Outlook 2007
- Microsoft Outlook 2010
- Thunderbird 8 and 9
- The iPhone (various iOS versions)
- Windows Phone 7

NOTE: You can download the email clients from the download section of Office 365.

These steps are explained with an example where the federated domain name is *namtest.com* and the base URL is *https://namtest.com:8443/nidp*. Replace the domain name and base URL based on your system configuration.

1. Open the Microsoft Online Services Module.
2. Run the following command:

   ```
   $cred=Get-Credential
   ```

   Specify your cloud service administrator account credentials.
3. Run the following command:

   ```
   Connect-MsolService –Credential $cred
   ```

   This cmdlet connects you to the cloud service.
4. Execute the following command to check the existing domain federation settings:

   ```
   Get-MsolDomainFederationSettings -DomainName namtest.com
   ```

   Substitute *namtest.com* with your domain name before executing this command.

   In the output, look for the `ActiveLogOnUri` parameter.

   For the Identity Server base URL `https://namtest.com:8443/nidp`, the value of the `ActiveLogOnUri` must be `https://namtest.com:8443/nidp/saml2/soap`. The `ActiveLogOnUri` is dependent on the base URL of the Identity Server.

   If the value of `ActiveLogOnUri` in the command output is `https://namtest.com:8443/nidp/saml2/soap`, go to Step 5 without modifying the configuration.

   (Conditional) If the `ActiveLogOnUri` is not `https://namtest.com:8443/nidp/saml2/soap`, execute the following command. Substitute *namtest.com* and port *8443* with your domain name and port number respectively before executing the following command.

   ```
   Set-MsolDomainFederationSettings -DomainName namtest.com -ActiveLogOnUri "https://
   namtest.com:8443/nidp/saml2/soap" -preferredauthenticationprotocol SAMLP
   ```
5. Create a new email account in your email client and enter your Office 365 email ID.

   NOTE: Configure Outlook related DNS settings before using email clients. You can configure these settings after adding the domain on the Office 365 port page.

   ---
6. The system prompts for specifying the basic authentication. Specify Access Manager credentials.

   The email account is created after successful authentication.

   NOTE: While logging in to the new email account, enter Access Manager credentials.

Verifying Single Sign-On Access

You need at least one user in Office 365 to verify that single sign-on is set up. If you have an existing user, ensure that the Immutable ID matches with the GUID of the Access Manager user.

Prerequisite:

- You need at least one user in Office 365 to verify that single sign-on is set up. If you have an existing user, ensure that the Immutable ID matches the GUID of the Access Manager user.

  For instance, if your user store is eDirectory and you want to retrieve the GUID of an existing Access Manager user, execute the following command on the eDirectory server terminal:

```
ldapsearch -D cn=<context> -w <password> -b <search base> cn=<name of the user>
GUID | grep GUID
```

Create an Office 365 user with this GUID as the Immutable ID using the following command in Powershell:

```
new-msolUser -userprincipalName "user1@domain name" -immutableID "GUID of
user1" - lastname "lastname of user 1" -firstname user1 -DisplayName "user1
users" -BlockCredential $false -"LicenseAssignment testdomain:ENTERPRISEPACK"
-usageLocation "two letter country code[example: US,IN,DE,BE,GB etc]" -Password
"password of the user".
```

Verifying Single Sign-on:

To verify that single sign-on is set up correctly, perform the following procedure in a server that is not added to the domain:

1. Go to Microsoft Online Services.
2. Log in with your corporate credentials. (For example : `user1@namtest.com`)

   If single sign-on is enabled, the password field is dimmed. You will instead see the following message: `You are now required to sign in at <your company>`.
3. Select the **Sign in at your company** link.

   If you are able to sign in without errors, single sign-on is set up successfully.

Useful Resources

The following list includes few useful resources for troubleshooting:

- Office 365 Troubleshooting
- Microsoft Remote Connectivity Analyzer
- Sign in to Office 365 for Business
- Description of Office 365 Desktop Setup Tool Logging Errors

Links for Troubleshooting Sign-in issues with Microsoft Lync:

- Describes how to troubleshoot sign-in issues with Microsoft Lync and Office 365
- Describes how to clear the credential cache that can occur on some operating systems
- Describes how Microsoft Lync can cache its connection endpoints and how to resolve the issue
- Describes how to troubleshoot common issues between Lync for Mac and Office 365

Troubleshooting Scenarios

- “WS-Trust and WS-Federation Scenarios” on page 586
- “SAML 2.0 Scenarios” on page 587
- “Office 365 Domain Scenarios” on page 588

WS-Trust and WS-Federation Scenarios

Issue in Setting Up a Domain for Federation

If you try to set a primary domain for federation by running the `Set-MsolDomainAuthentication` command, it throws the following error:

```
Set-MsolDomainAuthentication: You cannot remove this domain as the default domain
without replacing it with another default domain. Use the Set-MsolDomain cmdlet to
set another domain as the default domain before you delete this domain.
```

To fix this issue, change the default domain by performing the following steps:

1. In the Office 365 portal, click **Organization Name** on the Admin page.
2. Click **Edit**.
3. Select a new default domain.

Set-MsolDomainAuthentication : You cannot remove this domain as the default domain without replacing it with another default domain

If you get this error it indicates that you attempted to delete the default domain without replacing it with another domain.

Use the `Set-MsolDomain` cmdlet to set another domain as the default domain before you delete this domain.

After upgrading iOS Apps to the Latest Version, Single Sign-On to Office 365 Services Fail

To establish single sign-on from iOS apps to Office 365 services, perform the following steps:

1. In the Administration Console, click **Devices** > **Identity Servers** > **Edit** > **Local** > **Contract**.
2. Specify a name to identity the contract.
3. Specify the URI as `http://schemas.microsoft.com/ws/2008/06/identity/authenticationmethod/password`.
4. Select **Name/Password - Form - WebService** method.

SAML 2.0 Scenarios

- "SSO to MicroSoft Services Fails" on page 587
- "Issue in Setting Up a Domain for Federation" on page 587

SSO to MicroSoft Services Fails

SSO fails at Microsoft with this error:

```
Your organization could not sign you in to this service
```

Perform the following steps to fix this issue:

- Verify that the attributes are configured properly.

  You can also use the SAML tracer plug-in Firefox to review the SAML assertion sent to Office365.
- Verify that federation settings are using the `Get-MsolDomainFederationSettings -DomainName <YOUR DOMAIN>` command.

Issue in Setting Up a Domain for Federation

If you try setting up a primary domain for federation by running the `Set-MsolDomainAuthentication` command, it throws the following error:

```
Set-MsolDomainAuthentication: You cannot remove this domain as the default domain
without replacing it with another default domain. Use the Set-MsolDomain cmdlet to
set another domain as the default domain before you delete this domain.
```

To fix this issue, change the default domain by performing the following steps:

1. In the Office 365 portal, click **Organization Name** on the Admin page.
2. Click **Edit**.
3. Select a new default domain.

Office 365 Domain Scenarios

- "Issues with the Directory Synchronization Tool" on page 588
- "Active Profile Authentication Fails for Microsoft Exchange Clients" on page 588
- "Microsoft Online Services Sign-In Assistant Installation Fails If Microsoft Office Professional Plus Is Installed" on page 588
- "Single Sign-On to Office 365 Domain Fails" on page 588
- "No License to Use Office 365 Services" on page 589
- "After Initial Successful Authentication, Unending Loop While Logging into Lync Using Wrong Username and Password" on page 589

Issues with the Directory Synchronization Tool

- If the installation of the Directory Synchronization tool fails, check the Event Viewer. Installation may fail if the Microsoft Online Service Sign-In Assistant is already installed on the system.
- If you require to uninstall the Directory Synchronization tool, log off and then login.
- If the Directory Synchronization tool is slow, increase RAM of the server.

Active Profile Authentication Fails for Microsoft Exchange Clients

If the active profile authentication fails for Microsoft Exchange (Outlook) clients, verify that the necessary DNS records have been added to your DNS. For more information, see Create DNS records at any DNS hosting provider for Office 365.

Microsoft Online Services Sign-In Assistant Installation Fails If Microsoft Office Professional Plus Is Installed

Manually install Microsoft Online Services Sign-In Assistant, if its installation fails after installing Microsoft Office Professional Plus with this message:

```
"The Microsoft Online Services Sign In Assistant has experience an error. The error
must be resolved before your subscription for this product can be verified. To
retry subscription verification, first resolve error message 800704DD or try to
manually install the Microsoft Online Services Sign In Assistant...."
```

You can download the installer from MicroSoft Download Center.

After installation is complete, relaunch the service to verify your Office 365 license.

Single Sign-On to Office 365 Domain Fails

If single sign-on fails, ensure that the ImmutableID and the User Principal Name (UPN) matches the Office 365 user. To get Office 365 user details, log in to using Powershell and execute the following command:

```
Get-MsolUser -UserPrincipalName user1@namtest.com | fl *
```

No License to Use Office 365 Services

If you receive an error stating that the user does not have license to use Office365, Log in to Office 365 as an administrator and assign required service licenses to the user.

After Initial Successful Authentication, Unending Loop While Logging into Lync Using Wrong Username and Password

After successfully authenticating to Office 365 client, if you attempt to login to the Lync client using an incorrect username and password, Lync client uses the details from the previous successful session and tries to get a token from Access Manager. This results in an unending loop.

To resolve this issue, in the Lync client user interface, select the **Delete my sign-in info** option and log in once again.

Sample Tokens

- "Sample SAML Token" on page 589
- "Sample WS-Trust Token" on page 591
- "Sample WS-Federation Token" on page 593

Sample SAML Token

This section contains a sample XML for WS-Trust request and response.

Request:

```
<samlp:AuthnRequest xmlns:samlp="urn:oasis:names:tc:SAML:2.0:protocol"
xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion" ID="_3ae4edbc-7ab5-48c7-a08e-
b8d6e395e02c" IssueInstant="2012-09-09T08:41:35Z" Version="2.0"
AssertionConsumerServiceIndex="0" ><saml:Issuer>urn:federation:MicrosoftOnline</
saml:Issuer><samlp:NameIDPolicy Format="urn:oasis:names:tc:SAML:2.0:nameid-
format:persistent"/></samlp:AuthnRequest>
```

Response:

```
<samlp:Response xmlns:samlp="urn:oasis:names:tc:SAML:2.0:protocol"
xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion"
Consent="urn:oasis:names:tc:SAML:2.0:consent:obtained" Destination="https://
login.microsoftonline.com/login.srf" ID="idRuMHBvlVGqYUsw2Es-SbA5UeO8w"
InResponseTo="_3ae4edbc-7ab5-48c7-a08e-b8d6e395e02c" IssueInstant="2012-09-
09T08:41:51Z" Version="2.0">
  <saml:Issuer>https://www.netiqtst.com/nidp/saml2/metadata</saml:Issuer>
<samlp:Status>
    <samlp:StatusCode Value="urn:oasis:names:tc:SAML:2.0:status:Success"/>
</samlp:Status>
  <saml:Assertion ID="idF5JceWGWYwS3bOkmJS2wJuNqitU" IssueInstant="2012-09-
09T08:41:51Z" Version="2.0">
    <saml:Issuer>https://www.netiqtst.com/nidp/saml2/metadata</saml:Issuer>
<ds:Signature xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
      <ds:SignedInfo>
<CanonicalizationMethod xmlns="http://www.w3.org/2000/09/xmldsig#"
Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#"/>
<ds:SignatureMethod Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha1"/>
        <ds:Reference URI="#idF5JceWGWYwS3bOkmJS2wJuNqitU">
<ds:Transforms>
            <ds:Transform Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-
signature"/>
            <ds:Transform Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#"/>
```

```
          </ds:Transforms>
          <ds:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
          <DigestValue xmlns="http://www.w3.org/2000/09/
xmldsig#">ZocFiEUYcda0cKGRNcZYZqvmnlM=</DigestValue>
        </ds:Reference>
      </ds:SignedInfo>
      <SignatureValue xmlns="http://www.w3.org/2000/09/xmldsig#">
DLk4Uv/4VlwwKVz7XdDQOdUv8ltcryLv2U3K7q57AE70wk/NNsa4kP8Xdta36Y47Oj+XTV+a+q0y
YsMNIezySxaxMqo01Fm+6PfMH7HtTVj7fQ3n+VwANqbIs3G7eaaV1pHdUs79/dBujS8baNmlZEBR
2gGVMWCHOa1fTOSZO8yPt9ume0PsYXpo2RdaoGkJCZUnVIiIWg6UtI0zEKbY6mP3JhrUJ7OVHdbz
yNBzhfTv0m71nz0JKpy+i8MeDUIu1OiqTTIZ+c2SPceYhQcj8umrdE4JCGEBYNIE52Pa1bRYgmLd
roAKn56vLDjq04VnYVRGhqP/McZwYZrx+7E7qQ==
      </SignatureValue>
      <ds:KeyInfo>
        <ds:X509Data>
          <ds:X509Certificate>
MIIFBzCCA++gAwIBAgIRAKdqzGh19tecryvMuy+QhgAwDQYJKoZIhvcNAQEFBQAwcjELMAkGA1UE
BhMCR0IxGzAZBgNVBAgTEkdyZWF0ZXIgTWFuY2hlc3RlcjEQMA4GA1UEBxMHU2FsZm9yZDEaMBgG
A1UEChMRQ09NT0RPIENBIExpbWl0ZWQxGDAWBgNVBAMTD0Vzc2VudGlhbFNTTCBDQTAeFw0xMjA5
MDcwMDAwMDBaFw0xMjEyMDYyMzU5NTlaMFExITAfBgNVBAsTGERvbWFpbiBDb250cm9sIFZhbGlk
YXRlZDERMA8GA1UECxMIRnJlZSBTU0wxGTAXBgNVBAMTEHd3dy5uZXRpcXRzdC5jb20wggEiMA0G
CSqGSIb3DQEBAQUAA4IBDwAwggEKAoIBAQCX6k7wnFUoyPtqSj06xyQMHQtoxASBtHGASaOMxfZJ
rHQ4wbJUqMEtrxYcz4JxFrLzE8qvlY5r7cwxx/yvsiFwHq2HdRY6KU6I2u0eRF/tRwf3rl222/Xl
7wRbgdL43zd0yypjub9FKXlCxkaKucA1P+EVGTd7H8dFjMuf0iKZYvBFg9tcJWBGpFOw5iwe/rjK
6gQSXf13+Tpb6915lsusJfPMe3t04wA4XuyLlcJ/Jrxrj9xrEtwkmUcudTvEZRvJFnz3NYXcW0J8
6a0JZSEiHlVHrIY/44fVEQFjkrfr2u5RKGBJzl35xb2x5mkUSzzy4CSL5p0fCsVOve7LKx/fAgMB
AAGjggG3MIIBszAfBgNVHSMEGDAWgBTay+qtWwhdzP/8JlTOSeVVxjj0+DAdBgNVHQ4EFgQUEj/C
c5rqiBWiSzo9B8iJPdJnCpYwDgYDVR0PAQH/BAQDAgWgMAwGA1UdEwEB/wQCMAAwNAYDVR0lBC0w
KwYIKwYBBQUHAwEGCCsGAQUFBwMCBgorBgEEAYI3CgMDBglghkgBhvhCBAEwRQYDVR0gBD4wPDA6
BgsrBgEEAbIxAQICBzArMCkGCCsGAQUFBwIBFh1odHRwczovL3NlY3VyZS5jb21vZG8uY29tL0NQ
UzA7BgNVHR8ENDAyMDCgLqAshipodHRwOi8vY3JsLmNvbW9kb2NhLmNvbS9Fc3NlbnRpYWxTU0xD
QS5jcmwwbgYIKwYBBQUHAQEEYjBgMDgGCCsGAQUFBzAChixodHRwOi8vY3J0LmNvbW9kb2NhLmNv
bS9Fc3NlbnRpYWxTU0xDQV8yLmNydDAkBggrBgEFBQcwAYYYaHR0cDovL29jc3AuY29tb2RvY2Eu
Y29tMCkGA1UdEQQiMCCCEHd3dy5uZXRpcXRzdC5jb22CDG5ldGlxdHN0LmNvbTANBgkqhkiG9w0B
AQUFAAOCAQEAJoS/fE0gBMWvzQBsRRuSMBHMNbgDXP1fVPwJZnkfIHbb/wXwYK7AqA5efOe1Alqz
QD94kJ+W6JZm4ripePJk7QLnK2imqJb0E7LdmWQ3D05WQNsZKUklfR+9elP6xBN5ycXqtiEItScm
hE7H2gynz4/ejLXZv8XsBkfsYnT0wWUmyTsqYPLmVk7ELfPiPGZsQcvpmSO9eoTQ8zabkQGjquzM
NgGtXOMQBQgNO/7IMghgmSR0NduPguZoL31Ox84yKdf6Hl5cvbnH2W4c0n8vTkgCwUkB8ONY1Tge
6TFPwzS98PzV08nxKSJW1hckasLQAYcw++bC7Blz+Nc7YyrNPw==
          </ds:X509Certificate>
        </ds:X509Data>
      </ds:KeyInfo>
    </ds:Signature>
    <saml:Subject>
      <saml:NameID Format="urn:oasis:names:tc:SAML:2.0:nameid-format:persistent"
NameQualifier="https://www.netiqtst.com/nidp/saml2/metadata"
SPNameQualifier="urn:federation:MicrosoftOnline">bzM2NkBuZXRpcXRzdC5jb20=</
saml:NameID>
      <saml:SubjectConfirmation Method="urn:oasis:names:tc:SAML:2.0:cm:bearer">
        <saml:SubjectConfirmationData InResponseTo="_3ae4edbc-7ab5-48c7-a08e-
b8d6e395e02c" NotOnOrAfter="2012-09-09T09:41:51Z" Recipient="https://
login.microsoftonline.com/login.srf"/>
      </saml:SubjectConfirmation>
    </saml:Subject>
    <saml:Conditions NotBefore="2012-09-09T05:55:12Z" NotOnOrAfter="2012-09-
09T11:28:30Z">
      <saml:AudienceRestriction>
        <saml:Audience>urn:federation:MicrosoftOnline</saml:Audience>
      </saml:AudienceRestriction>
    </saml:Conditions>
```

```
    <saml:AuthnStatement AuthnInstant="2012-09-09T08:41:51Z"
SessionIndex="idF5JceWGWYwS3bOkmJS2wJuNqitU">
      <saml:AuthnContext>

<saml:AuthnContextClassRef>urn:oasis:names:tc:SAML:2.0:ac:classes:Password</
saml:AuthnContextClassRef>
       <saml:AuthnContextDeclRef>urn:oasis:names:tc:SAML:2.0:ac:classes:Password</
saml:AuthnContextDeclRef>
      </saml:AuthnContext>
    </saml:AuthnStatement>
    <saml:AttributeStatement>
     <saml:Attribute xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http:/
/www.w3.org/2001/XMLSchema-instance" Name="IDPEmail"
NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:unspecified">
        <saml:AttributeValue xsi:type="xs:string">o3662@netiqtst.com</
saml:AttributeValue>
      </saml:Attribute>
     <saml:Attribute xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http:/
/www.w3.org/2001/XMLSchema-instance" Name="ImmutableID"
NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:unspecified">
        <saml:AttributeValue xsi:type="xs:string">bzM2NkBuZXRpcXRzdC5jb20=</
saml:AttributeValue>
      </saml:Attribute>
    </saml:AttributeStatement>
  </saml:Assertion>
</samlp:Response>
```

Sample WS-Trust Token

```
<saml:Assertion AssertionID="nsts150b8594-0aff-424f-8113-46045d943171"
IssueInstant="2014-05-09T07:00:18.019Z" Issuer="https://namnetiq.in/nidp/wsfed/"
MajorVersion="1" MinorVersion="1" xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
xmlns:exc14n="http://www.w3.org/2001/10/xml-exc-c14n#"
xmlns:saml="urn:oasis:names:tc:SAML:1.0:assertion" xmlns:xs="http://www.w3.org/
2001/XMLSchema">
   <saml:Conditions NotBefore="2014-05-09T07:00:18.019Z" NotOnOrAfter="2014-05-
09T07:06:18.019Z">
      <saml:AudienceRestrictionCondition>
         <saml:Audience>
          urn:federation:MicrosoftOnline
         </saml:Audience>
      </saml:AudienceRestrictionCondition>
   </saml:Conditions>
   <saml:Advice/>
   <saml:AuthenticationStatement AuthenticationInstant="2014-05-09T07:00:18.019Z"
AuthenticationMethod="urn:oasis:names:tc:SAML:1.0:am:password">
      <saml:Subject>
         <saml:NameIdentifier Format="urn:oasis:names:tc:SAML:1.1:nameid-
format:unspecified" NameQualifier="urn:federation:MicrosoftOnline">
          TLP1nEzIc0EEtEyz9ZxMyA==
         </saml:NameIdentifier>
         <saml:SubjectConfirmation>
            <saml:ConfirmationMethod>
             urn:oasis:names:tc:SAML:1.0:cm:bearer
            </saml:ConfirmationMethod>
         </saml:SubjectConfirmation>
      </saml:Subject>
   </saml:AuthenticationStatement>
   <saml:AttributeStatement>
      <saml:Subject>
```

```
         <saml:NameIdentifier Format="urn:oasis:names:tc:SAML:1.1:nameid-
format:unspecified" NameQualifier="urn:federation:MicrosoftOnline">
          TLP1nEzIc0EEtEyz9ZxMyA==
         </saml:NameIdentifier>
         <saml:SubjectConfirmation>
            <saml:ConfirmationMethod>
             urn:oasis:names:tc:SAML:1.0:cm:bearer
            </saml:ConfirmationMethod>
         </saml:SubjectConfirmation>
      </saml:Subject>
      <saml:Attribute AttributeName="UPN" AttributeNamespace="http://
schemas.xmlsoap.org/claims">
         <saml:AttributeValue xmlns:xs="http://www.w3.org/2001/XMLSchema">
          namtest@namnetiq.in
         </saml:AttributeValue>
      </saml:Attribute>
      <saml:Attribute AttributeName="ImmutableID" AttributeNamespace="http://
schemas.microsoft.com/LiveID/Federation/2008/05">
         <saml:AttributeValue xmlns:xs="http://www.w3.org/2001/XMLSchema">
          TLP1nEzIc0EEtEyz9ZxMyA==
         </saml:AttributeValue>
      </saml:Attribute>
   </saml:AttributeStatement>
   <ds:Signature xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
      <ds:SignedInfo>
         <ds:CanonicalizationMethod Algorithm="http://www.w3.org/2001/10/xml-exc-
c14n#"/>
         <ds:SignatureMethod Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-
sha1"/>
         <ds:Reference URI="#nsts150b8594-0aff-424f-8113-46045d943171">
            <ds:Transforms>
               <ds:Transform Algorithm="http://www.w3.org/2000/09/
xmldsig#enveloped-signature"/>
               <ds:Transform Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#"/>
            </ds:Transforms>
            <ds:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
            <ds:DigestValue>
             0Zvo3DbV0Qq7m9q7ER4Hol24bmA=
            </ds:DigestValue>
         </ds:Reference>
      </ds:SignedInfo>
      <ds:SignatureValue>
       SqWAA39fYb3VJPBebZ6bsiUh0C+8ElbgDv2yG6xq3WLYUX/DoQ6RLfsb/1mVmMQBcGqhUxhcDRAT
k6JA3djHbZCrZh7qblc8uBr+nm1SzpS/BO7todTLu+g835WGSKdpnpSoTjhO285MjsoomnrL+A4S
33F5Ld5OVOTPoar1wpBPFOgm7k9SnzjU0h7yIpP7YlzX1uF2sPvNeDRhkNEIsWwSPUY9mw04An9V
AsC1Cb1Q7+vEtCxggJ4A6nxk8G9bvPRisk7H5fTihf0THNEzu5s6KnyGHCc6k2/jWHHF4Appg/aJ
Ze1yQR9MKagNe60sAU2U83GM8WUst+o3+PvI3A==
      </ds:SignatureValue>
      <ds:KeyInfo>
         <ds:X509Data>
            <ds:X509Certificate>

MIIFSTCCBDGgAwIBAgIGb+MI39nZMA0GCSqGSIb3DQEBCwUAMIHGMQswCQYDVQQGEwJVUzEQMA4G
A1UECBMHQXJpem9uYTETMBEGA1UEBxMKU2NvdHRzZGFsZTElMCMGA1UEChMcU3RhcmZpZWxkIFRl
Y2hub2xvZ2llcywgSW5jLjEzMDEGA1UECxMqaHR0cDovL2NlcnRzLnN0YXJmaWVsZHRlY2guY29t
L3JlcG9zaXRvcnkvMTQwMgYDVQQDEytTdGFyZmllbGQgU2VjdXJlIENlcnRpZmljYXRlIEF1dGhv
cml0eSAtIEcyMB4XDTE0MDUwNjA5MDYwNVoXDTE1MDIyNjEyMDQwNFowOTEhMB8GA1UECxMYRG9t
YWluIENvbnRyb2wgVmFsaWRhdGVkMRQwEgYDVQQDEwtuYW1uZXRpcS5pbjCCASIwDQYJKoZIhvcN
AQEBBQADggEPADCCAQoCggEBAMzjEinlOiwzMpKBQO+H2sb+HifrmVi7JDzhRfOKJakG+nXsgVx2
QRToN0UbvoeqlDtaTZSKrFb0mc/E3aEkgSU67DAzWvtm3nUSboJc4QVWQlJmXIP989K2H1DastwE
```

```
Srg6iw0MMUuz9ZadP3BQjV4VVB9qX81D32lD4Ti1gJYUDg5tpaUnftddiR+rZQROea3ABC0+oeZa
7w+jVFUOAP+uG2iJ4zksIO+F3wIXDNZMYQwFlTvnCTO6/4cRW1XoGxh0BbZGdYn0qHzAOu9okT2B
gnz+aTaMGSIPpPr+PXjB31XqeAhBRoXgrddWIt1DawyrJETPOrzfMhd1i+QSXHcCAwEAAaOCAccw
ggHDMAwGA1UdEwEB/wQCMAAwHQYDVR0lBBYwFAYIKwYBBQUHAwEGCCsGAQUFBwMCMA4GA1UdDwEB
/wQEAwIFoDA7BgNVHR8ENDAyMDCgLqAshipodHRwOi8vY3JsLnN0YXJmaWVsZHRlY2guY29tL3Nm
aWcyczEtOC5jcmwwWQYDVR0gBFIwUDBOBgtghkgBhv1uAQcXATA/MD0GCCsGAQUFBwIBFjFodHRw
Oi8vY2VydGlmaWNhdGVzLnN0YXJmaWVsZHRlY2guY29tL3JlcG9zaXRvcnkvMIGCBggrBgEFBQcB
AQR2MHQwKgYIKwYBBQUHMAGGHmh0dHA6Ly9vY3NwLnN0YXJmaWVsZHRlY2guY29tLzBGBggrBgEF
BQcwAoY6aHR0cDovL2NlcnRpZmljYXRlcy5zdGFyZmllbGR0ZWNoLmNvbS9yZXBvc2l0b3J5L3Nm
aWcyLmNydDAfBgNVHSMEGDAWgBQlRYFoUCY4PTstLL7Natm2PbNmYzAnBgNVHREEIDAeggtuYW1u
ZXRpcS5pboIPd3d3Lm5hbW5ldGlxLmluMB0GA1UdDgQWBBQANClvlYFFU3cAkvFQz/TxuttEUTAN
BgkqhkiG9w0BAQsFAAOCAQEAySHcxqGpgrm9HSiSIFzDOdC9BraZdjh+fIUBeKRUBmSjSByPJIHj
OGuBnY8FtuPY8/e1KhzwhZcuUhY3zwVQzbWStWLraySJyO1SzRRJC4onLbx42ARdKbRgxA/JDsmY
aTnyYq+ZOLm6XUtDweFEDkklAy2sO8gru54ogJ0iD/JyX/dgZEH/v9lGjdNFUDwG4dLz++a2Ol/U
UfqJye7Rb5UgNkewcG9KjydiTgP7Mv6m8/JjzOl31ejIVVqwz30fo+agirrIWWG2Ogtk0JUFrY73
coKTzspPszxMGN2FJpRSymtO+cqVlEuAK6/SCr2mhBvxg4GJuXuzSLp2kSrIfA==
           </ds:X509Certificate>
         </ds:X509Data>
      </ds:KeyInfo>
   </ds:Signature>
</saml:Assertion>
```

Sample WS-Federation Token

```
<wst:RequestedSecurityToken xmlns:wst="http://schemas.xmlsoap.org/ws/2005/02/
trust">
      <saml:Assertion AssertionID="idjTptEEQd5CuKy-0M-MBCY9lDHVQ"
IssueInstant="2014-05-09T06:44:07Z" Issuer="https://namnetiq.in/nidp/wsfed/"
MajorVersion="1" MinorVersion="1"
xmlns:saml="urn:oasis:names:tc:SAML:1.0:assertion">
         <saml:Conditions NotBefore="2014-05-09T06:29:07Z" NotOnOrAfter="2014-05-
09T06:59:07Z">
            <saml:AudienceRestrictionCondition>
               <saml:Audience>
                urn:federation:MicrosoftOnline
               </saml:Audience>
            </saml:AudienceRestrictionCondition>
         </saml:Conditions>
        <saml:AuthenticationStatement AuthenticationInstant="2014-05-09T06:44:07Z"
AuthenticationMethod="name/password/uri">
            <saml:Subject>
               <saml:NameIdentifier Format="urn:oasis:names:tc:SAML:1.1:nameid-
format:unspecified">
                TLP1nEzIc0EEtEyz9ZxMyA==
               </saml:NameIdentifier>
               <saml:SubjectConfirmation>
                  <saml:ConfirmationMethod>
                   urn:oasis:names:tc:SAML:1.0:cm:bearer
                  </saml:ConfirmationMethod>
               </saml:SubjectConfirmation>
            </saml:Subject>
         </saml:AuthenticationStatement>
         <saml:AttributeStatement>
            <saml:Subject>
               <saml:NameIdentifier Format="urn:oasis:names:tc:SAML:1.1:nameid-
format:unspecified">
                TLP1nEzIc0EEtEyz9ZxMyA==
               </saml:NameIdentifier>
               <saml:SubjectConfirmation>
                  <saml:ConfirmationMethod>
```

```
                  urn:oasis:names:tc:SAML:1.0:cm:bearer
                </saml:ConfirmationMethod>
             </saml:SubjectConfirmation>
          </saml:Subject>
          <saml:Attribute AttributeName="UPN" AttributeNamespace="http://
schemas.xmlsoap.org/claims">
             <saml:AttributeValue>
              XX
             </saml:AttributeValue>
          </saml:Attribute>
          <saml:Attribute AttributeName="ImmutableID" AttributeNamespace="http:/
/schemas.microsoft.com/LiveID/Federation/2008/05">
             <saml:AttributeValue>
              XX
             </saml:AttributeValue>
          </saml:Attribute>
       </saml:AttributeStatement>
       <ds:Signature xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
          <ds:SignedInfo xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
             <CanonicalizationMethod Algorithm="http://www.w3.org/2001/10/xml-
exc-c14n#" xmlns="http://www.w3.org/2000/09/xmldsig#"/>
             <ds:SignatureMethod Algorithm="http://www.w3.org/2000/09/
xmldsig#rsa-sha1" xmlns:ds="http://www.w3.org/2000/09/xmldsig#"/>
             <ds:Reference URI="#idjTptEEQd5CuKy-0M-MBCY9lDHVQ" xmlns:ds="http:/
/www.w3.org/2000/09/xmldsig#">
                <ds:Transforms xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
                   <ds:Transform Algorithm="http://www.w3.org/2000/09/
xmldsig#enveloped-signature" xmlns:ds="http://www.w3.org/2000/09/xmldsig#"/>
                   <ds:Transform Algorithm="http://www.w3.org/2001/10/xml-exc-
c14n#" xmlns:ds="http://www.w3.org/2000/09/xmldsig#"/>
                </ds:Transforms>
                <ds:DigestMethod Algorithm="http://www.w3.org/2000/09/
xmldsig#sha1" xmlns:ds="http://www.w3.org/2000/09/xmldsig#"/>
                <DigestValue xmlns="http://www.w3.org/2000/09/xmldsig#">
                 vOVgMA5UmoGFqXL4ENvYPsH/aP0=
                </DigestValue>
             </ds:Reference>
          </ds:SignedInfo>
          <SignatureValue xmlns="http://www.w3.org/2000/09/xmldsig#">

hwPIdSGG+M29sih+5MiWEf862d5K/zSST3XVn1kIwWN3HaLi/yAnGiOUf6nzNJxE99pudElUdy3R
Kc5z8iQAu3gekVG1Nk4n2mDKZVet1kKEcgHGsfdwGxCkz5bpsPsaMB+pJyvFqu/RlRXIqsZtVrxv
7PwOIwUPxJQesNhJrdoJNsKxr65ckj2EeL5scCrDh9mYvtMCh/Qa0C3ALXUm+hBfj21hqw1Qp58I
m68DFTwh35pDkm4AXVxSRCm/9FKuoPGSXeU+O016Gv/FISLiEma+48dN0awlJvxzPI/cUayyJU2N
3EZp7LpZLfErushLBQQ9YmDNmevpCQoN4cZtuA==

          </SignatureValue>
          <ds:KeyInfo xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
             <ds:X509Data xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
               <ds:X509Certificate xmlns:ds="http://www.w3.org/2000/09/xmldsig#">

MIIFSTCCBDGgAwIBAgIGb+MI39nZMA0GCSqGSIb3DQEBCwUAMIHGMQswCQYDVQQGEwJVUzEQMA4G
A1UECBMHQXJpem9uYTETMBEGA1UEBxMKU2NvdHRzZGFsZTElMCMGA1UEChMcU3RhcmZpZWxkIFRl
Y2hub2xvZ2llcywgSW5jLjEzMDEGA1UECxMqaHR0cDovL2NlcnRzLnN0YXJmaWVsZHRlY2guY29t
L3JlcG9zaXRvcnkvMTQwMgYDVQQDEytTdGFyZmllbGQgU2VjdXJlIENlcnRpZmljYXRlIEF1dGhv
cml0eSAtIEcyMB4XDTE0MDUwNjA5MDYwNVoXDTE1MDIyNjEyMDQwNFowOTEhMB8GA1UECxMYRG9t
YWluIENvbnRyb2wgVmFsaWRhdGVkMRQwEgYDVQQDEwtuYW1uZXRpcS5pbjCCASIwDQYJKoZIhvcN
AQEBBQADggEPADCCAQoCggEBAMzjEinlOiwzMpKBQO+H2sb+HifrmVi7JDzhRfOKJakG+nXsgVx2
QRToN0UbvoeqlDtaTZSKrFb0mc/E3aEkgSU67DAzWvtm3nUSboJc4QVWQlJmXIP989K2H1DastwE
Srg6iw0MMUuz9ZadP3BQjV4VVB9qX81D32lD4Ti1gJYUDg5tpaUnftddiR+rZQROea3ABC0+oeZa
```

```
7w+jVFUOAP+uG2iJ4zksIO+F3wIXDNZMYQwFlTvnCTO6/4cRW1XoGxh0BbZGdYn0qHzAOu9okT2B
gnz+aTaMGSIPpPr+PXjB31XqeAhBRoXgrddWIt1DawyrJETPOrzfMhd1i+QSXHcCAwEAAaOCAccw
ggHDMAwGA1UdEwEB/wQCMAAwHQYDVR0lBBYwFAYIKwYBBQUHAwEGCCsGAQUFBwMCMA4GA1UdDwEB
/wQEAwIFoDA7BgNVHR8ENDAyMDCgLqAshipodHRwOi8vY3JsLnN0YXJmaWVsZHRlY2guY29tL3Nm
aWcyczEtOC5jcmwwWQYDVR0gBFIwUDBOBgtghkgBhv1uAQcXATA/MD0GCCsGAQUFBwIBFjFodHRw
Oi8vY2VydGlmaWNhdGVzLnN0YXJmaWVsZHRlY2guY29tL3JlcG9zaXRvcnkvMIGCBggrBgEFBQcB
AQR2MHQwKgYIKwYBBQUHMAGGHmh0dHA6Ly9vY3NwLnN0YXJmaWVsZHRlY2guY29tLzBGBggrBgEF
BQcwAoY6aHR0cDovL2NlcnRpZmljYXRlcy5zdGFyZmllbGR0ZWNoLmNvbS9yZXBvc2l0b3J5L3Nm
aWcyLmNydDAfBgNVHSMEGDAWgBQlRYFoUCY4PTstLL7Natm2PbNmYzAnBgNVHREEIDAeggtuYW1u
ZXRpcS5pboIPd3d3Lm5hbW5ldGlxLmluMB0GA1UdDgQWBBQANClvlYFFU3cAkvFQz/TxuttEUTAN
BgkqhkiG9w0BAQsFAAOCAQEAySHcxqGpgrm9HSiSIFzDOdC9BraZdjh+fIUBeKRUBmSjSByPJIHj
OGuBnY8FtuPY8/e1KhzwhZcuUhY3zwVQzbWStWLraySJyO1SzRRJC4onLbx42ARdKbRgxA/JDsmY
aTnyYq+ZOLm6XUtDweFEDkklAy2sO8gru54ogJ0iD/JyX/dgZEH/v9lGjdNFUDwG4dLz++a2Ol/U
UfqJye7Rb5UgNkewcG9KjydiTgP7Mv6m8/JjzOl31ejIVVqwz30fo+agirrIWWG2Ogtk0JUFrY73
coKTzspPszxMGN2FJpRSymtO+cqVlEuAK6/SCr2mhBvxg4GJuXuzSLp2kSrIfA==

                </ds:X509Certificate>
             </ds:X509Data>
          </ds:KeyInfo>
       </ds:Signature>
    </saml:Assertion>
 </wst:RequestedSecurityToken>
```

6 Access Manager Policies

Policies provide the authorization component of Access Manager Appliance. The administrator of the Identity Server can use policies to define how properties of a user's authenticated identity map to the set of active roles for the user. This role definition serves as the starting point for role-based authorization policies of the Access Gateway. Additionally, you can define authorization policies to control access to protected resources based on user and system attributes other than assigned roles.

Policies are very flexible. You can, for example, set up a policy that permits or denies access to a protected Web site, depending on user roles (such as employee or manager), the value of an LDAP attribute, or the user's IP address.

The Access Gateway includes an Embedded Service Provider agent that interacts with the Identity Server to provide authentication, policy decision, and policy enforcement. For Web application servers, the Access Gateway provides the ability to inject the user's roles into HTTP headers to allow integration with the Web server's authorization processes.

This section describes how Access Manager uses policies to assign roles to control access and to enable single sign-on to resources that require credentials. Topics include:

- "Understanding Policies" on page 591
- "Role Policies" on page 603
- "Authorization Policies" on page 642
- "Identity Injection Policies" on page 691
- "Form Fill Policies" on page 711
- "External Attribute Source Policies" on page 739
- "Risk-Based Policies" on page 745

Understanding Policies

Policies are logical and testable rules to maintain security and consistency within your Access Manager Appliance infrastructure. You can specify activation criteria, deactivation criteria, temporal constraints (such as time of day or subnet), identity constraints (such as user object attribute values), and additional separation-of-duty constraints. Identity information can come from any identity source (such LDAP, an Identity Vault, or a directory) or from the Access Manager's Identity Server, which provides full Liberty Alliance specification support and SAML 2.0 support. Identity is available throughout the determination of rights and permissions.

- "Selecting a Policy Type" on page 592
- "Tuning the Policy Performance" on page 592
- "Managing Policies" on page 593
- "Managing Policy Containers" on page 595
- "Managing a Rule List" on page 596
- "Adding Policy Extensions" on page 598
- "Enabling Policy Logging" on page 602

Selecting a Policy Type

Access Manager Appliance uses the policy type to define the context within which a policy is evaluated. Each type of policy differs in purpose, which in turn determines the conditions and actions that apply. For example, the conditions and actions of an Authorization policy differ from the conditions and actions of an Identity Injection policy.

When you click **New** on the Policies page, the system displays the predefined policy types in a drop-down list. Each policy type represents the set of conditions and actions that are available. You then configure rules to determine user roles, make decision requests, and enforce authorization decisions. You can also set up policies with no conditions, allowing actions to always take place. As policies and conditions become complex, it can be simpler and more manageable to design policies with conditions that deny or restrict access to large groups of users, rather than setting up policies that permit access to certain users.

Access Manager Appliance has the following policy types:

- **Access Gateway: Authorization:** This policy type is used to permit or deny access to protected resources, such as Web servers. After you have set up the protected resource, you use the policy rules to define how you want to restrict access. For example, if a user is denied access to a resource, you can use the policy to redirect them to a URL where they can request access to the resource.
- **Access Gateway: Identity Injection:** This policy type evaluates the rules for Identity Injection, which retrieves identity data from a data source (user store) and forwards it to Web applications. Such a policy can enable single sign-on. After the user has authenticated, the policy supplies the information required by the resource rather than allowing the resource to prompt the user for the information.
- **Access Gateway: Form Fill:** This policy type creates a policy that automatically fills in the information required in a form, after the form is filled the first time. Use this policy to configure single sign-on for resources that require form data and for injecting JavaScript to an HTML page. You can also use this policy for injecting JavaScript to HTML pages.
- **Identity Server: Roles:** This policy type evaluates rules for establishing the roles of an authenticated user. Roles are generated based on policy statements each time a user authenticates. Roles are placed into an Authentication Profile, which can be used as input in policies for Authorization or Identity Injection.
- **Identity Server: External Attribute Source:** This policy type is used to create a policy that retrieves the attributes from external sources.

Tuning the Policy Performance

Authorization and Identity Injection policies allow you to select conditions, one of which is Roles. If you have thousands of users accessing your resources, you might want to design most of your policies to use roles. Roles are evaluated when a user logs in, and the roles assigned to the user are cached as long as the session is active. When the user accesses a resource protected by a policy that uses role conditions, the policy can be immediately evaluated because the user's role values are available. This is not true for all conditions; the values for some conditions must be retrieved from the

user store. For example, if the policy uses a condition with an LDAP attribute, the user's value must be retrieved from the LDAP user store before the policy can be evaluated. On a system with medium traffic, this delay is not noticed. On a system with high traffic, the delay might be noticeable.

However, you can design your policies to have the same results without retrieving the LDAP attribute value at resource access. You can create a Role policy for the LDAP attribute and have users assigned to this role at authentication when they match the attribute value requirements. When users access resources, they gain immediate access or are immediately denied access because their role assignments are cached.

If the same LDAP attribute policy is used to grant access to multiple resources, chances that a user notices a delay are minimal. The first time a policy is evaluated for a user, the data required for the policy is cached and is therefore immediately available the next time it is requested.

Another option available for LDAP, Credential Profile, Liberty User Profile, and Shared Secret attributes is to have the attribute values sent with the assertion at authentication. You configure an attribute set for the attributes, and then configure the service provider for these attributes. For more information, see "Configuring the Attributes Sent with Authentication" on page 156.

As you design your policies, experiment and find the type that works best for your network and your customers.

Managing Policies

1. In the Administration Console, click **Policies** > **Policies**.
2. In the Policy Container drop-down list, select the container.

   If you have not created any containers, only the Master_Container is available in the list.
3. You can perform the following tasks from this page:
   - "Creating Policies" on page 593
   - "Sorting Policies" on page 594
   - "Deleting Policies" on page 594
   - "Renaming or Copying a Policy" on page 594
   - "Importing and Exporting Policies" on page 594
   - "Refreshing Policy Assignments" on page 594

Creating Policies

Before creating policies, you need to design your policy strategy. For example, if you are going to use role-based access, you need to decide which roles you need and which roles allow access to your protected resources. Roles, which are used by Authorization policies that grant and deny access, need to be created first. If you have already created the roles and assigned them to users in your LDAP user store, you can use the values of your role attributes in the Authorization policies rather than using Access Manager Appliance roles.

To create a policy, see the following sections:

- Chapter , "Role Policies," on page 603
- Chapter , "Authorization Policies," on page 642
- Chapter , "Identity Injection Policies," on page 691
- Chapter , "Form Fill Policies," on page 711
- Chapter , "External Attribute Source Policies," on page 739

Sorting Policies

Policies can be sorted by name and by type. On the Policies page, click **Name** in the **Policy List**, and the policies are sorted alphabetically by name. To sort alphabetically by type, click **Type** in the **Policy List**.

You can also use containers to organize your policies. For more information, see “Managing Policy Containers” on page 595.

Deleting Policies

A policy cannot be deleted as long as a resource is configured to use the policy. This means that you must remove the policy from all protected resources for the Access Gateway.

Roles can be used by Authorization, Form Fill, and Identity Injection policies. Before you can delete a Role policy, you must remove any reference to the role from all other policies.

Renaming or Copying a Policy

Copy: To copy a policy, select a policy, click **Copy**, then click **OK**. The new policy is named “Copy of ...” This is useful when you are creating multiple policies that require only minor variations to make them unique. You must rename the policy after making these modifications.

Rename: To rename a policy, select a policy, click **Rename**, specify a new name, then click **OK**.

Importing and Exporting Policies

Policies that are created in the Administration Console can be exported and used in another Administration Console that is managing a different group of Access Gateways and other devices. Each policy type has slightly different import requirements. See the following:

- “Importing and Exporting Role Policies” on page 642
- “Importing and Exporting Authorization Policies” on page 691
- “Importing and Exporting Identity Injection Policies” on page 709
- “Importing and Exporting Form Fill Policies” on page 734

Refreshing Policy Assignments

If you have made changes in policy assignments that are not reflected on the page, click **Refresh References**. This action can take a while to complete if you have numerous policies and have assigned them to protect numerous resources. The Administration Console needs to verify the configuration of each device.

If you have made changes in policy assignments that are not reflected on the page, click **Refresh References**. This action can take a while to complete if you have numerous policies and have assigned them to protect numerous resources. The Administration Console needs to verify the configuration of each device.

Viewing Policy Information

The **Policy List** table displays the following information about each policy:

Column	Description
Name	Displays the name of the policy. To modify a policy, click its name.
Type	Specifies the type of policy (Authorization, Identity Injection, Roles, or Form Fill) and the type of resource that can use it (Identity Server or Access Gateway).
Used By	Displays the name of the Access Gateway or the Identity Server configuration that the policy is assigned to. If the policy is unassigned, this column has no value. If the policy is assigned to a protected resource, click the down-arrow button to view the names of the resources it has been assigned to.
Extensions Used	Specifies whether the policy uses any extensions. If none has been used, this column has no value.
Description	Displays a description of the policy. If no description has been specified, this column has no value.

Managing Policy Containers

You use policy containers to store and organize policies, similar to how you organize files in folders. The **Master_Container** is a permanent policy container, but you can use the **Containers** tab to create new containers.

A policy container can hold up to 500 policies. When you reach that limit, you must create another container to add, copy, or import policies. For performance and for ease in finding a policy, you might want to limit a container to 200 or fewer policies. Policies in a container can be sorted by name and type, to aid you in finding a particular policy.

If you have only one administrator configuring and managing policies, you can create additional policy containers to help you keep policies organized. If you have multiple administrators creating policies, you can create a container for each administrator to use. This allows multiple administrators to modify policies at the same time. When an administrator opens a policy in a container, the container is locked, which prevents other administrators from modifying any policies in that container until changes are applied or canceled.

1. In the Administration Console, click **Policies** >**Containers**.
2. On the Containers page, click **New**.
3. Name the policy container, then click **OK**.
4. Click **Close**.

   After you add a policy container, the system displays it in the **Policy Container** drop-down list on the Policy List page.

You must delete all the policies in a policy container before you can delete the policy container.

1. Select the check box of the policy container. Click **Delete**.A Confirm dialog box displays a message "Number of Containers selected: 1, Delete selected Containers?"
2. Click **Ok** to continue or **Cancel** to close the window

Managing a Rule List

You configure rules to create a policy. The rules collectively represent a desired course of action when the required conditions are met, such as denying entry-level employees access to a secure Web site, and permitting access for employees who have a role of Manager.

When the system evaluates the policy conditions, it begins with the rule with the highest priority and evaluates the conditions, starting with the first condition group in the rule. Each rule contains one or more conditions and one or more actions. If a rule's conditions are met, the rule's action is performed. For some policy types, the performance of any rule's action terminates the policy evaluation. With Authorization policies, for example, after the policy has determined that a user is either permitted or denied access to a resource, there is no reason to evaluate the policy further. However, a Role policy might identify multiple roles to which a user belongs. In this case, each rule of the policy must be evaluated to determine all roles to which the user belongs.

IMPORTANT: The interface for the policy engine is designed for flexibility. It does not protect you from creating rules that do nothing because they are always true or always false. For example, you can set up a condition where Client IP is equal to Client IP, which is always true. You are responsible for defining the condition so that it does a meaningful comparison.

To manage the list of rules for a policy:

1. In the Administration Console, click **Policies > Policies**.
2. Select the container.
3. Click the name of the policy.
4. In the **Rule List** section, select one of the following:

   New: To create a new rule, click **New**.

   You use multiple rules to coordinate how a policy operates, and the behavior varies according to the policy type. To understand how multiple rules are evaluated, see the following:

   - "Rule Evaluation for Role Policies" on page 596
   - "Rule Evaluation for Authorization Policies" on page 597
   - "Rule Evaluation for Identity Injection and Form Fill Policies" on page 597

   Delete: Select a rule, then click this option to delete the rule. If the policy has only one rule, you cannot delete the last rule.

   Copy: Select a rule, then click this option to copy a rule. To modify the copy, click the rule number.

   Enable: Select a rule, then click this option to enable a rule.

   Disable: Select a rule, then click this option to disable a rule.
5. Click **OK**, then click **Apply Changes**.

Rule Evaluation for Role Policies

A Role policy is used to determine which role or roles a user is assigned to. However, you can specify only one role per rule. Role policies are evaluated when a user authenticates. Role policies do not directly deny or allow access to any resource, nor do they determine if a user is authenticated. A user's role can be used in the evaluation of an Authorization policy, but at that point the evaluation of the role policy has already occurred and is not directly part of the authorization process. The performance of an action (assigning a user to a role) does not terminate the evaluation of the policy, so subsequent rules in the policy continue to be evaluated.

Rule Evaluation for Authorization Policies

When the Access Gateway discovers a rule in an Authorization policy that either permits or denies a user access to a protected resource, it stops processing the rules in the policy. Use the following guidelines in determining whether your Authorization policy needs multiple rules:

- If the policy enforces multiple access requirements that can result in differing actions (either permit or deny), use separate rules to define the conditions and actions.
- If you want other conditions or actions processed when a rule fails, you must create a second rule for the users that fail to match the conditions.

If you create multiple rules, you can modify the order that the rules are processed. This allows you to create policies that contain a number of Permit rules that allow access if the user matches the rule. The lowest priority rule in such a policy is a Deny rule, which denies access to everyone who has not previously matched a Permit rule.

IMPORTANT: If you create policies with multiple Permit rules, you must make the last rule in the policy a generic deny policy (a rule with no conditions and with an action of deny). This ensures that if the Result on Error Condition field in a rule is set incorrectly, the user matches the last rule and is denied access. Without this rule, a user might gain access because the user didn't match any of the rules.

You can also create a number of policies and enable multiple policies for the same protected resource. Rule priority determines how the enabled policies interact with each other. The rules in the policies are gathered into one list, then sorted by priority. The processing rules are applied as if the rules came from one policy. It is a personal design issue whether you create a policy with multiple rules or create multiple policies that you enable on a single protected resource. Either design produces a list of rules, sorted by priority, that is applied to the user requesting access to the protected resource.

Rule Evaluation for Identity Injection and Form Fill Policies

Rules in Identity Injection and Form Fill policies have actions, but no conditions. Because they have no conditions, all the rules are evaluated and the actions are performed. Identity Injection policies have two exceptions to this rule; they can insert only one authentication header and one cookie header. If you create multiple rules, each with an authentication header and a cookie header, the rule with the highest priority is processed and its actions performed. The actions in the second rule for injecting an authentication header and a cookie header are ignored.

You cannot create multiple rules for a Form Fill policy.

Viewing Rules

The policy view administrators can view the information related to rules.The rules collectively represent a desired course of action when the required conditions are met, such as denying entry-level employees access to a secure Web site, and permitting access for employees who have a role of Manager.

When the system evaluates the policy conditions, it begins with the rule with the highest priority and evaluates the conditions, starting with the first condition group in the rule. Each rule contains one or more conditions and one or more actions. If a rule's conditions are met, the rule's action is performed. For some policy types, the performance of any rule's action terminates the policy evaluation.

To view the list of rules for a policy:

1. In the Administration Console, click **Policies > Policies**.
2. Select the container.
3. Click the name of the policy.
4. In the **Rule List** section, the policy view administrator can view the following details.:

   Type: Displays the type of the policy.

   Description: Displays the description of the policy type.

   Priority: Displays the priority of the rule number to the policy type.

   Actions: Displays the actions for the policy type.

Adding Policy Extensions

If Access Manager Appliance does not supply the action, the data type, or the condition that you need for a policy, you can add a customized policy extension. For example, suppose you need a policy that permits access based on whether a user has a specific role which is assigned to users in an Oracle database. The custom extension could read the role assignments of the user from the Oracle database and return a string containing the role names. This data could then be used to determine access rights to Access Manager resources.

For information about how to create a policy extension, see the *NetIQ Access Manager 4.2 Developer Guide*.

After a policy extension has been created, you need to perform the following tasks to use the extension:

- “Installing the Extension on the Administration Console” on page 598
- “Distributing a Policy Extension” on page 600

After you have configured the extension, you can perform the following tasks:

- “Managing a Policy Extension Configuration” on page 601
- “Viewing Extension Details” on page 602

Installing the Extension on the Administration Console

The policy extension can be delivered as either a `.jar` file or a `.zip` file.

- “Uploading and Configuring a JAR File” on page 598
- “Importing a ZIP File” on page 600

Uploading and Configuring a JAR File

To install an extension, you need to have access to the `.jar` file and know the following information about the extension or extensions contained within the file:

What you need to create	◆ A display name for the extension. ◆ A description for the extension.

What you need to know	◆ The policy type of the extension, which defines the policy type it can be used with. You must know whether it is an extension for an Access Gateway Authorization policy, an Access Gateway Identity Injection policy, or an Identity Server Role policy. ◆ The name of the Java class that is used by the extension. Each data type usually uses a different Java factory class. ◆ The filename of the extension. ◆ The names, IDs, and mapping type of any configuration parameters. Configuration parameters allow the policy engine to pass data to the extension, which the extension can then use to retrieve data or to evaluate a condition. ◆ The type of data the extension manipulates. **Authorization Policy:** Can be used to return the following: ◆ An action of deny, permit, or obligation. ◆ A condition that the extension evaluates and returns either true or false. ◆ A data element that the extension retrieves and the policy can use for evaluating a condition. **Identity Injection Policy:** A data extension that retrieves data for injecting into a header. **Identity Role Policy:** Can be used to return the following: ◆ A condition that the extension evaluates and returns either true or false. ◆ A data element that the extension retrieves which can be used in evaluating a condition or used to assign roles. **External Attribute Source Policy:** A data extension that retrieves attributes from external sources.

If the file contains more than one extension, you need to create a configuration for each extension in the file.

1. Copy the `.jar` file to a location that you can browse to from the Administration Console.
2. In the Administration Console, click **Policies** > **Extensions**.
3. To upload the file, click **Upload** > **Browse**, select the file, then click **Open**.
4. (Conditional) If you want this `.jar` file to overwrite an existing version of the file, select **Overwrite existing *.jar file**.
5. Click **OK**.

   The file is uploaded to the Administration Console, but nothing is visible on the Extensions page until you create a configuration.
6. To create an extension configuration, click **New**, then fill in the following fields:

   Name: Specify a display name for the extension.

   Description: (Optional) Specify the purpose of the extension and how it must be used.

   Policy Type: From the drop-down list, select the type of extension you have uploaded.

   Type: From the drop-down list, select the data type of the extension.

   Class Name: Specify the name of the class that creates the extension, such as com.acme.policy.action.successActionFactory.

File Name: From the drop-down list, select the `.jar` file that contains the Java class that implements the extension and its corresponding factory. This must be the file you uploaded in Step 3.

7 Click **OK**.

8 (Conditional) If the extension requires data from Access Manager Appliance, click the name of the extension.

9 In the **Configuration Parameters** section, click **New**, specify a name and ID, then click **OK**.

The developer of the extension must supply the name and ID that the extension requires.

10 In the **Mapping** column, click the down-arrow, then select the required data type.

The developer of the extension must supply the data type that is required. If the data type is a data string, then the developer needs to explain the type of information you need to supply in the text field.

11 (Conditional) If the extension requires more than one data item, repeat Step 9 and Step 10.

12 Click **OK**.

The extension is now available for the policy type it was created for.

13 (Conditional) If the class can be used for multiple policy types, you need to create an extension configuration for each policy type.

For example, if an extension can be used for both an Identity Injection policy and a Role policy, you need to create an entry for both. The **File Name** option must contain the same value, but the other options must contain unique values.

14 Continue with "Distributing a Policy Extension" on page 600.

Importing a ZIP File

A `.zip` file with an exported extension contains both the `.jar` file and the extension configuration.

1 Copy the `.zip` file to a location that you can browse to from the Administration Console.

2 In the Administration Console, click **Policies** > **Extensions**.

3 To upload the file, click **Upload** > **Browse**, select the file, then click **Open**.

4 (Conditional) If you want the `.jar` file in the import to overwrite an existing version of the file, select **Overwrite existing *.jar file**.

5 Click **OK**.

The extension is imported in the Administration Console.

6 (Conditional) If the extension requires some customizing, click the name of the extension and follow the instructions that came with the extension.

7 Continue with "Distributing a Policy Extension" on page 600.

Distributing a Policy Extension

To distributed the policy extension to the devices that need it:

1 Create a policy that uses the extension:

- **Role Policy:** To create a Role policy that uses the extension, see "Creating Roles" on page 608.
- **Identity Injection Policy:** To create an Identity Injection policy that uses the extension, see "Configuring an Identity Injection Policy" on page 693.

- **Authorization Policy:** To create an Authorization policy that uses the extension, see “Creating Access Gateway Authorization Policies” on page 653.
- **External Attribute Source Policy:** To create an External Attribute Source policy that uses the extension, see Chapter , “External Attribute Source Policies,” on page 739.

2 Assign the policy to a device:

- For a Role policy, enable it for an Identity Server.

  For more information, see “Enabling and Disabling Role Policies” on page 642.
- For an Authorization policy, assign it to a protected resource.

  For more information, see “Assigning an Authorization Policy to a Protected Resource” on page 107.
- For an Identity Injection policy, assign it to a protected resource.

  For more information, see “Assigning an Identity Injection Policy to a Protected Resource” on page 107.
- For an External Attribute Source policy, enable it for an Identity Server.

  For more information, see “Enabling External Attributes Policy” on page 739.

IMPORTANT: Do not update the device at this time. The `.jar` files must be distributed before you update the device.

---

3 Distribute the `.jar` files:

3a Click **Policies** > **Extensions**.

3b Select the extension, then click **Distribute JARs**.

3c Restart Tomcat on the devices listed for reboot.

- **Linux:** Enter the following commands:

  In the Access Gateways: `/etc/init.d/novell-mag restart`.

  In the Identity Servers: `/etc/init.d/novell-idp restart`.
- **Windows:** Enter the following commands:

```
net stop Tomcat7

net start Tomcat7
```

4 (Conditional) If the extension is for an Authorization policy or an Identity Injection policy, update the Access Gateway.

Managing a Policy Extension Configuration

1 In the Administration Console, click **Policies** > **Extensions**.

2 To export a policy extension, select the policy, then click **Export**.

3 To delete an extension, a policy cannot be using it. Use the **Used By** column to determine the policies that are using the extension. Modify the listed policies. When the extension is no longer used by any policies, select the extension, then click **Delete**.

4 To rename a policy extension, select the extension, click **Rename**, specify a new name, then click **OK**. When a policy extension is renamed and the extension is in use by a policy, the policy is updated. This causes the **Apply Changes** button to be active on the **Policy List** page.

Viewing Extension Details

You can modify the details of an existing extension and control the information Access Manager Appliance provides to the extension when the data is evaluated.

1. In the Administration Console, click **Policies** > **Extensions**.
2. Click the name of the extension.

   You can view or modify the following details:

   Description: (Optional) Specifies the purpose of the extension and how it must be used.

   Class Name: Specifies the name of the class that creates the extension, for example com.acme.policy.action.successActionFactory.

   File Name: Specifies the `.jar` file that contains the Java class that implements the extension and its corresponding factory. Select the appropriate file from the drop-down list.
3. (Conditional) Specify the Condition Parameters required by the extension.

   The documentation for the extension must tell you the number of parameters it requires and the data type of each parameter. You create the name and ID for the parameter, and they need to be unique for the extension.

   - To add a configuration parameter, click **New**, enter a name (a string) and an ID (a number) for the parameter, then click **OK**. In the **Mapping** field, click the down-arrow, then select the data item from the list. The selected data is available whenever the extension class is called to evaluate an action, a condition, or data.
   - To delete a configuration parameter, select the parameter, then click **Delete**.
4. Click **OK**.

Enabling Policy Logging

Policy logging is expensive; it uses processing time and disk space. In a production environment, you must enable it only under the following types of conditions:

- You have created a new policy and need to verify its functionality.
- You are troubleshooting a policy that is not behaving as expected.

To gather troubleshooting information, you must enable the **File Logging** and **Echo To Console** options in the Identity Server configuration and set the **Component File Logger Levels** for **Application** to at least **info**. Then you must update the Identity Server configuration and restart any Access Gateway Embedded Service Providers, so that the Embedded Service Providers read the logging options. See “Configuring Logging for Identity Server” on page 852. When you have solved the problem, you must disable these options.

The log file on the component that executed the policy is where you must look for logging information. For example, if you have an Access Gateway: Authorization error, look at the log on the Access Gateway that executed the policy.

For additional policy troubleshooting procedures, see Chapter , “Troubleshooting Access Manager Policies,” on page 1004.

Role Policies

This section describes the following topics for the Identity Server roles.

- “Understanding RBAC in Access Manager Appliance” on page 603
- “Enabling Role-Based Access Control” on page 607
- “Creating Roles” on page 608
- “Example Role Policies” on page 627
- “Creating Access Manager Appliance Roles in an Existing Role-Based Policy System” on page 630
- “Mapping Roles between Trusted Providers” on page 640
- “Enabling and Disabling Role Policies” on page 642
- “Importing and Exporting Role Policies” on page 642

Understanding RBAC in Access Manager Appliance

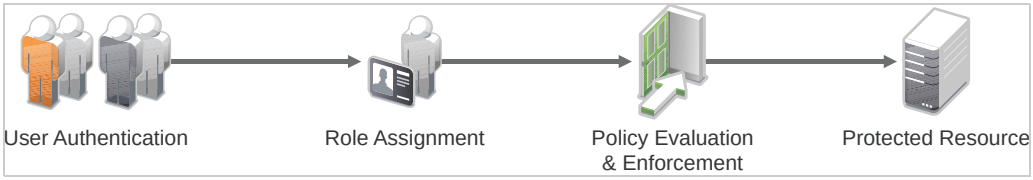
Role-based access control (RBAC) provides a convenient way to assign a user to a particular job function or set of permissions within an enterprise, in order to control access. As an administrator, you probably have defined a set of roles for your needs. Your roles might include Employee, Student, Administrator, Manager, and so on. You might have Web resources that you want available to all employees, or only to managers, as shown in Figure 6-1.

Figure 6-1 *Traditional RBAC*



Access Manager Appliance supports core RBAC functionality by providing user role mapping and the mapping of roles to resource rights and permissions. User role mapping is a primary function of a Role policy. Role mapping to resource rights is accomplished through Authorization policies. When creating a role, you assign users to the role, based on attributes of their identities. You also specify the constraints to place on the role.

Figure 6-2 *RBAC Using a Policy*



As shown in Figure 6-2, during user authentication, the system checks the existing Role policy to determine which roles that a user must be assigned to. After authentication, assigned roles can be used as evaluated conditions of an Authorization policy.

Web server applications can also be configured to use roles for access control. For these applications you can use Access Manager Appliance to assign the users to the required roles. You can use the Access Gateway Identity Injection policies to inject the assigned roles into the HTTP header that is sent to the Web server.

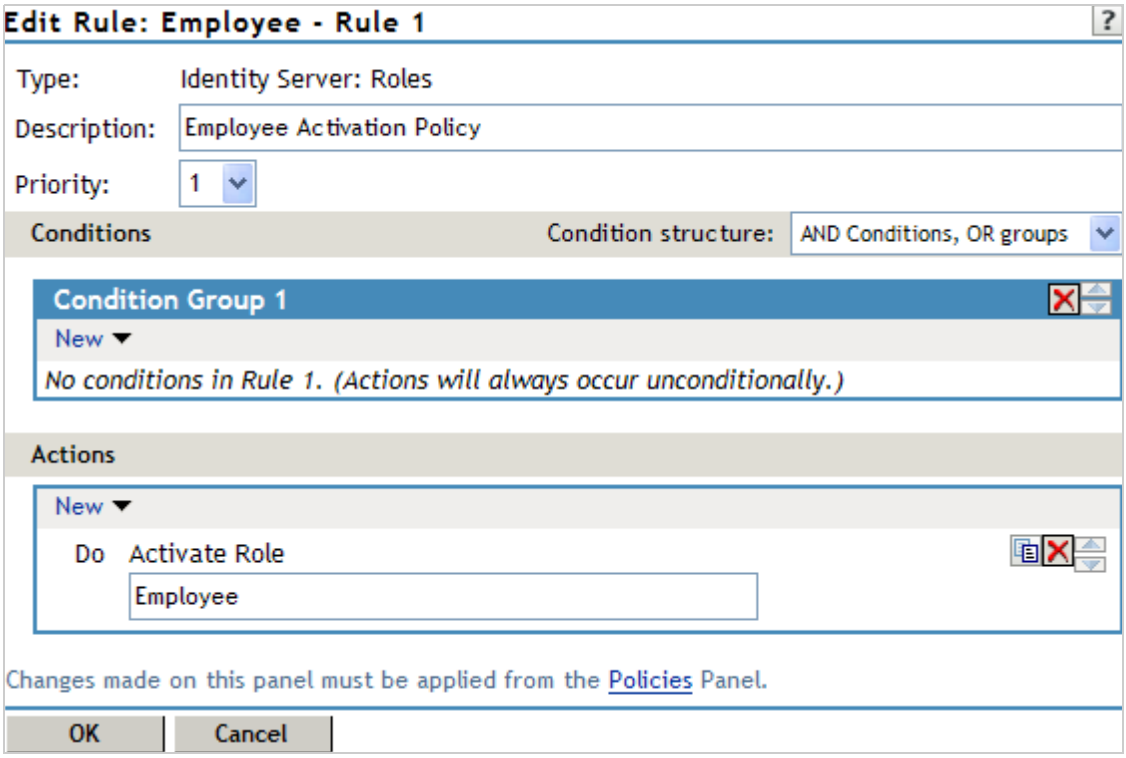
The following examples describe ways to use roles in Access Manager Appliance:

- “Assigning All Authenticated Users to a Role” on page 604
- “Using a Role to Create an Authentication Policy” on page 604
- “Using Prioritized Rules in an Authorization Policy” on page 607

Assigning All Authenticated Users to a Role

The system assigns users to roles when they authenticate. The following example illustrates a Role policy that creates an Employee role. All authenticated users are assigned to the role of Employee, because it does not include any conditions (see “Creating an Employee Role” on page 628).

Figure 6-3 *Employee Role Policy*



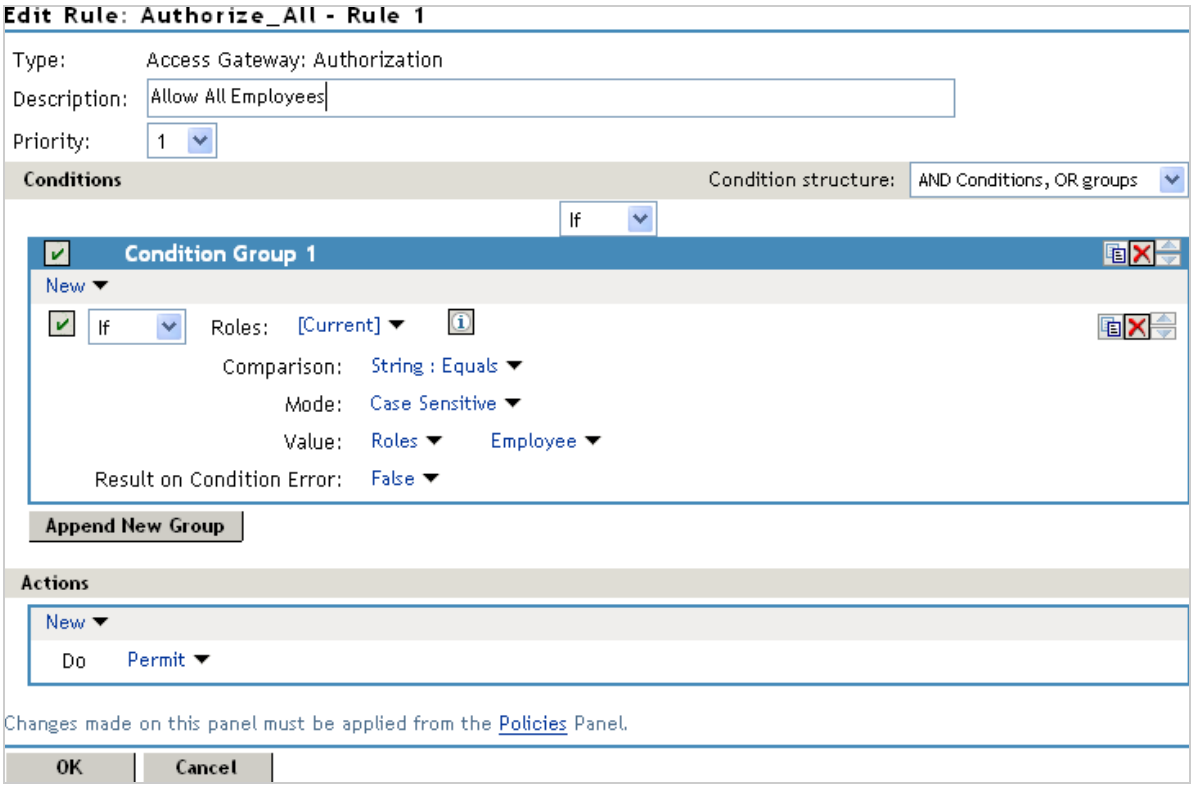
Role assignment audit events can be created during authentication to the Identity Server. You enabled this on the Logging page in the Identity Server configuration when you enable the **Login Provided** or **Login Consumed** options.

Using a Role to Create an Authentication Policy

The simplest implementation of RBAC policies is to include roles as evaluated conditions when creating Authorization policies.

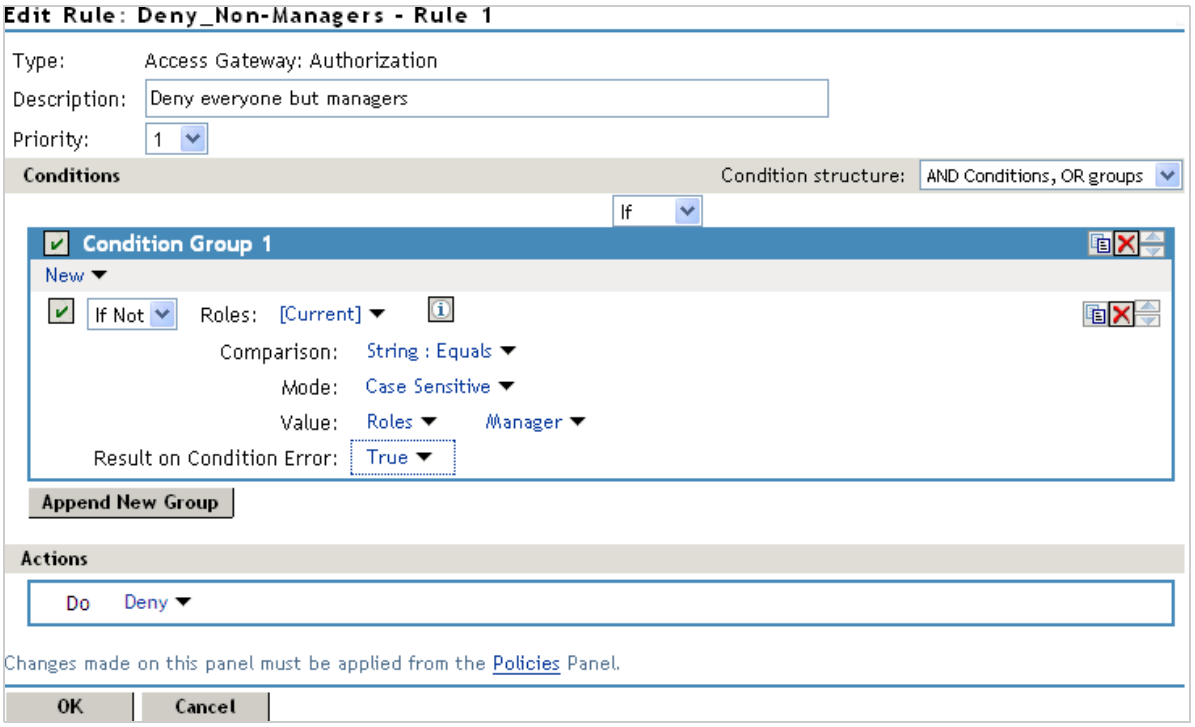
Suppose you belong to a company of 300 employees, and ten of them are managers. You can assign all employees to an Employee role, and make it a condition of an Authorization policy with no restrictions. Such a policy would permit access to Web resources intended for all employees, as shown in the following example:

Figure 6-4 *Employee Authorization Policy*



For more sensitive Web resources intended only for managers, you might create a role called Manager. (See “Creating a Manager Role” on page 628). The Manager role might be a condition of an Authorization policy that denies access to any employee that has not been assigned to the Manager role when the user authenticated. The following example illustrates this. Notice that the operand for the governing condition logic is set to `If Not`.

Figure 6-5 *Manager Authorization Policy*



After you have created the Authorization policies, you need to assign the policies to the resources they were designed to protect.

See “Assigning an Authorization Policy to a Protected Resource” on page 107.

Using Prioritized Rules in an Authorization Policy

In another policy example, you might create an Authorization policy for the Sales Department and set up a list of rules that evaluate whether a user has been assigned to one of the roles associated with the department, and then deny access if the user has not been assigned to any of them.

The following image illustrates this scenario:

Figure 6-6 *Authorization Policy with Multiple Rules*



In this example, you specify a first-priority rule with a condition that allows access if a user has been assigned to the role of Sales Representative. You add rules for users assigned to the a role of Sales Manager, Sales Vice President, and so on. You then create a lowest-priority rule that contains no conditions, and an action of Deny. This policy denies any user who has not been assigned a Sales department role. When users do not meet the conditions of the rules, the user is denied access by the lowest-priority rule.

For more information about using roles in Authorization policies, see Chapter , “Authorization Policies,” on page 642.

Enabling Role-Based Access Control

Role-based access control is used to provide a convenient way to assign a user to a particular job function or set of permissions within an enterprise, in order to control access. In Access Manager, you assign users to roles, based on attributes of their identity, and then associate policies to the role.

To assign a role to users at authentication, you must enable it for the Identity Server configuration.

1. In the Administration Console, click **Devices > Identity Servers > Servers > Edit > Roles**.
2. Click the role policy’s check box, then click **Enable**.
3. To disable the role policy, click the role policy’s check box, then click **Disable**.
4. To create a new role, click **Manage Policies**.
5. After enabling or disabling role policies, update the Identity Server configuration on the **Servers** tab.

Creating Roles

To implement RBAC, you must first define all of the roles within your organization and the permissions attached to each role. A collection of users requiring the same access can be assigned to a single role. Each user can also be assigned to one or more roles and receive the collective rights associated with the assigned roles. A role policy consists of one or more rules, and each rule consists of one or more conditions and an action.

1. In the Administration Console, click **Policies > Policies**.
2. Select the policy container, then click **New**.
3. Specify a name for the policy, then select **Identity Server: Roles** for the type of policy.
4. Fill in the following fields:

   Description: (Optional) Describe the purpose of this rule. If your role policy contains multiple rules, use the description to identify the purpose of each rule.

   Priority: Specify the order in which a rule is applied in the policy, when the policy has multiple rules. The highest priority is 1 and 10 is the lowest.
5. To create a condition for a policy rule, click **New** in the **Condition Group 1** section, then select one of the following:
   - **Authenticating IDP:** Specifies the identity provider that authenticated the current user. To use this condition, you must have set up a trusted relationship with more than one identity provider. For configuration information, see Authenticating IDP Condition.
   - **Authentication Contract:** Specifies the contract used to authenticate the current user. The selections in this list are defined in the Identity Server configuration. For configuration information, see Authentication Contract Condition.
   - **Authentication Method:** Specifies the method used to authenticate the current user. For configuration information, see Authentication Method Condition.
   - **Authentication Type:** Compares a selected authentication type to the authentication types used to authenticate the current user. For configuration information, see Authentication Type Condition.
   - **Credential Profile:** Requires the user to use the specified credential for authentication. Only values used at authentication time are available for this comparison. For configuration information, see Credential Profile Condition.
   - **LDAP Group:** Specifies a group in which the authenticating user is evaluated for membership. For configuration information, see LDAP Group Condition.
   - **LDAP OU:** Specifies an OU against which the authenticating user's container is evaluated for containment. For configuration information, see LDAP OU Condition.
   - **LDAP Attribute:** Specifies an attribute from the user object of an authenticated user. By default, the selection values include those defined for the InetOrgPerson class. For configuration information, see LDAP Attribute Condition.
   - **Liberty User Profile:** Specifies any one of a number of data values that have been mapped to a Liberty Profile attribute. For configuration information, see Liberty User Profile Condition.
   - **Roles from Identity Provider:** Specifies a role that has been assigned to the user by an identity provider. For configuration information, see Roles from Identity Provider Condition.
   - **User Store:** Compares a selected user store to the user store where the current user is authenticated. For configuration information, see User Store Condition.

- **Virtual Attribute:** Specifies a virtual attribute. The virtual attribute is used to store the transformed attribute values in the user's session. For configuration information, see "Virtual Attribute Condition" on page 623
- **Condition Extension:** (Conditional) If you have loaded and configured a role condition extension, this option specifies a condition that is evaluated by an outside source. See the documentation that came with the extension for information about what is evaluated.
- **Data Extension:** (Conditional) If you have loaded and configured a role data extension, this option specifies the value that the extension retrieves. You can then select to compare this value with an LDAP attribute, a Liberty User Profile attribute, a Data Entry Field, or another Data Extension. For more information, see the documentation that came with the extension.

NOTE: To improve the policy's performance, configure the LDAP Attributes, Credential Profile, and Liberty User Profile attributes to be sent with authentication. For more information, see"Configuring the Attributes Sent with Authentication" on page 156.

---

6 (Conditional) To add multiple conditions, repeat Step 5.

For more information about using multiple conditions in a rule, see "Using Multiple Conditions" on page 624.

7 In the **Actions** section, select one of the following:

- **Activate Role:** Select this option to specify a name for the role. If you are creating a role that needs to be injected into an HTTP header, use the capitalization format that the Web server expects.
- **Activate Selected Role:** Select this option to obtain the role value from an external source.

For more information about specifying a role or roles to activate, see "Selecting an Action" on page 625.

8 Click **OK** twice.

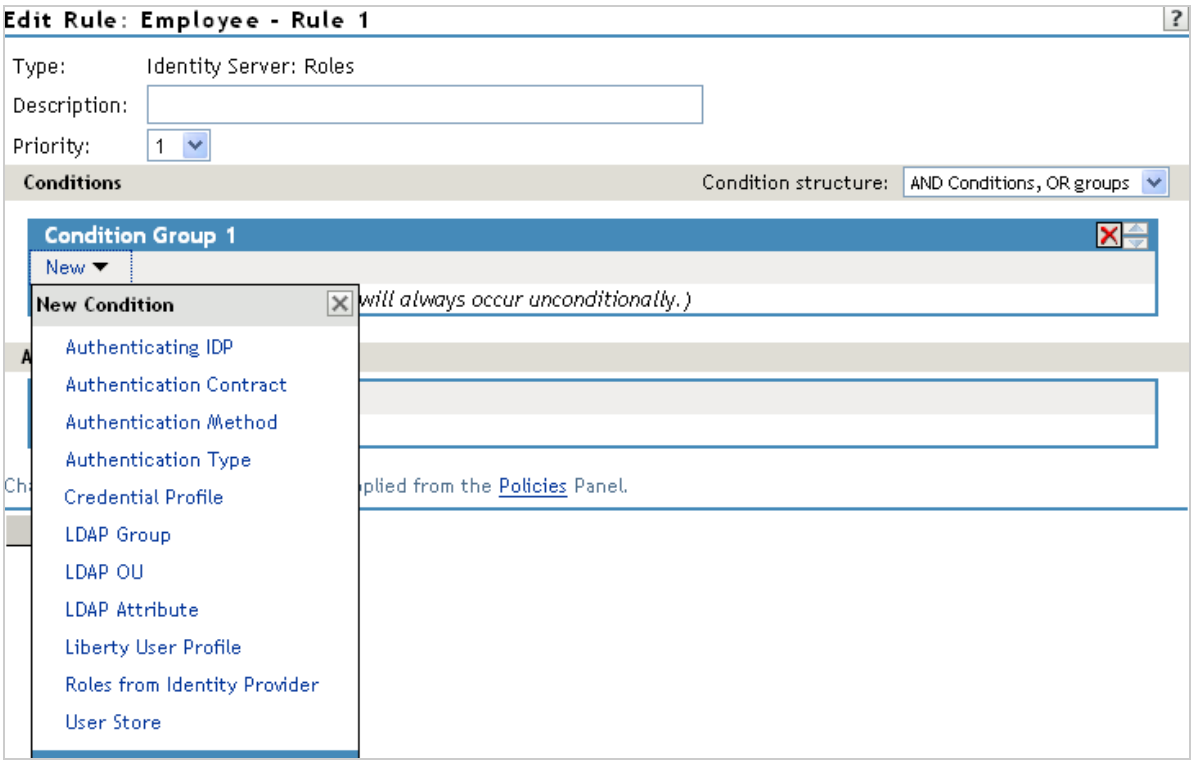
9 Click **Apply Changes**.

10 To enable the role for an Identity Server configuration, see "Enabling and Disabling Role Policies" on page 642.

Selecting Conditions

You create a role by selecting the appropriate conditions that qualify a user to be assigned to a role, as shown in the following image:

Figure 6-7 *Role Policy Conditions*



The following sections describe the conditions available for a Role policy:

- “Authenticating IDP Condition” on page 611
- “Authentication Contract Condition” on page 612
- “Authentication Method Condition” on page 614
- “Authentication Type Condition” on page 615
- “Credential Profile Condition” on page 616
- “LDAP Group Condition” on page 617
- “LDAP OU Condition” on page 618
- “LDAP Attribute Condition” on page 619
- “Liberty User Profile Condition” on page 620
- “Roles from Identity Provider Condition” on page 621
- “User Store Condition” on page 622
- “Virtual Attribute Condition” on page 623
- “Condition Extension” on page 623
- “Data Extension” on page 623

Authenticating IDP Condition

The Authenticating IDP condition allows you to assign a role based on the identity provider that authenticated the current user. To use this condition, you must have set up a trusted relationship with more than one identity provider. See “Managing Trusted Providers” on page 150.

The most common way to use this condition is when you have a service provider that has been configured to trust two identity providers and you want to assign a role based on which identity provider authenticated the user. To configure such a policy:

- Set the Authenticating IDP field to **[Current]**
- Set the **Value** field to Authenticating IDP
- Select the name of an identity provider

For the condition to evaluate to True, the identity provider specified in the policy must be the one that the user selected for authentication.

Comparison: Specify how the contract is compared to the data in the **Value** field. Select either a string comparison or a regular expression:

- **Comparison: String:** Specifies that you want the values compared as strings and how you want the string values compared. Select one of the following:
  - **Equals:** Indicates that the values must match, letter for letter.
  - **Starts with:** Indicates that the Authenticating IDP value must begin with the letters specified in the **Value** field.
  - **Ends with:** Indicates that the Authenticating IDP value must end with the letters specified in the **Value** field.
  - **Contains Substring:** Indicates that the Authenticating IDP value must contain the letters, in the same sequence, as specified in the **Value** field.
- **Comparison: Regular Expression: Matches:** Specifies that you want the values compared as regular expressions.

Mode: Select the mode appropriate for the comparison type:

- **Comparison: String:** Specify whether case is important by selecting **Case Sensitive** or **Case Insensitive**.
- **Comparison: Regular Expression: Matches:** Select one or more of the following:

  Canonical Equivalence
  Case Insensitive
  Comments
  Dot All
  Multi-Line
  Unicode
  Unix Lines

  For regular expression syntax information, see the Javadoc for java.util.regex.Pattern.

Value: Specify the value you want to compare with the Authenticating IDP value. If you select a static value for the Authenticating IDP value, select **Authenticating IDP** and **Current**. If you select **Current** for the Authenticating IDP value, select **Authenticating IDP**, then select the name of an identity provider.

Other value types are possible if you selected **Current** for the Authenticating IDP value. Your policy requirements determine whether they are useful.

Result on Condition Error: Specify what the condition returns when the comparison of the two values returns an error rather than the results of the comparison. Select either **False** or **True**. If you do not want the action applied when an error occurs, select **False**. If you want the action applied when an error occurs, select **True**.

Authentication Contract Condition

The Authentication Contract allows you to assign a role based on the contract the user used for authentication. The Identity Server has the following default contracts:

Name	URI
Name/Password - Basic	basic/name/password/uri
Name/Password - Form	name/password/uri
Secure Name/Password - Basic	secure/basic/name/password/uri
Secure Name/Password - Form	secure/name/password/uri

To configure other contracts for your system, click **Devices** > **Identity Servers** > **Edit** > **Local** > **Contracts**.

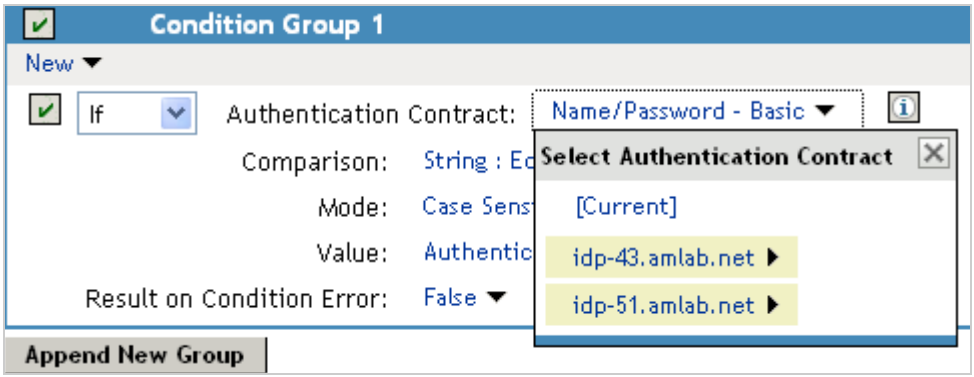
The most common way to use this condition is to select **[Current]** for the **Authentication Contract** field and to select **Authentication Contract** and the name of a contract for the **Value** field.

To specify an Authentication Contract condition, fill in the following fields:

Authentication Contract: To compare the contract that the user used with a static value, select **Current**. To compare a static value with what the user used, select a contract from the list.

If you have created more than one Identity Server configuration, select the configuration, then select the contract. The name of the contract is displayed. When you select this name, the configurations that contain a definition for this contract are highlighted.

For example, the following policy has selected **Name/Password - Basic** as the contract.



Two Identity Server configurations have been defined (idp-43.amlab.net and idp-51.amlab.net). Both configurations are highlighted because **Name/Password - Basic** is a contract that is automatically defined for all Identity Server configurations.

If the contract you are selecting for a condition is a contract with ORed credentials, you need to use multiple conditions to set up a rule. See "Creating a Rule for a Contract with ORed Credentials" on page 629.

Comparison: Specify how the contract is compared to the data in the **Value** field. Select either a string comparison or a regular expression:

- **Comparison: String:** Specifies that you want the values compared as strings and how you want the string values compared. Select one of the following:
  - **Equals:** Indicates that the values must match, letter for letter.
  - **Starts with:** Indicates that the Authentication Contract value must begin with the letters specified in the **Value** field.
  - **Ends with:** Indicates that the Authentication Contract value must end with the letters specified in the **Value** field.
  - **Contains Substring:** Indicates that the Authentication Contract value must contain the letters, in the same sequence, as specified in the **Value** field.
- **Comparison: Regular Expression: Matches:** Specifies that you want the values compared as regular expressions.

Mode: Select the mode appropriate for the comparison type:

- **Comparison: String:** Specify whether case is important by selecting **Case Sensitive** or **Case Insensitive**.
- **Comparison: Regular Expression: Matches:** Select one or more of the following:

  Canonical Equivalence
  Case Insensitive
  Comments
  Dot All
  Multi-Line
  Unicode
  Unix Lines

  For regular expression syntax information, see the Javadoc for java.util.regex.Pattern.

Value: Specify the value you want to compare with the Authentication Contract value. If you select a static value for the Authentication Contract value, select **Authentication Contract** and **Current**. If you select **Current** for the Authentication Contract value, select **Authentication Contract**, then select the name of a contract.

Other value types are possible if you selected **Current** for the Authentication Contract value. For example:

- You can select **Data Entry Field**. The value specified in the text box must be the URI of the contract for the conditions to match. For a list of these values, click **Devices** > **Identity Servers** > **Edit** > **Local** > **Contracts**.
- If you have defined a Liberty User Profile attribute for URI of the authentication contract, you can select **Liberty User Profile,** then select the attribute.
- If you have defined an LDAP attribute for URI of the authentication contract, you can select **LDAP Attribute,** then select the attribute.

Result on Condition Error: Specify what the condition returns when the comparison of the two values returns an error rather than the results of the comparison. Select either **False** or **True**. If you do not want the action applied when an error occurs, select **False**. If you want the action applied when an error occurs, select **True**.

Authentication Method Condition

The Authentication Method allows you to assign a role based on the method the user used for authentication.

Authentication Method: To compare the method that the user used with a static value, select **Current**. To compare a static value with what the user used, select a method from the list.

If you have created more than one Identity Server configuration, select the configuration, then select the method. The name of the method is displayed. When you select this name, the configurations that contain a definition for this method are highlighted.

Comparison: Specify how the method is compared to the data in the **Value** field. Select either a string comparison or a regular expression:

- **Comparison: String:** Specifies that you want the values compared as strings and how you want the string values compared. Select one of the following:
  - **Equals:** Indicates that the values must match, letter for letter.
  - **Starts with:** Indicates that the Authentication Method value must begin with the letters specified in the **Value** field.
  - **Ends with:** Indicates that the Authentication Method value must end with the letters specified in the **Value** field.
  - **Contains Substring:** Indicates that the Authentication Method value must contain the letters, in the same sequence, as specified in the **Value** field.
- **Comparison: Regular Expression: Matches:** Specifies that you want the values compared as regular expressions.

Mode: Select the mode appropriate for the comparison type:

- **Comparison: String:** Specify whether case is important by selecting **Case Sensitive** or **Case Insensitive**.
- **Comparison: Regular Expression: Matches:** Select one or more of the following:

  Canonical Equivalence
  Case Insensitive
  Comments
  Dot All
  Multi-Line
  Unicode
  Unix Lines

  For regular expression syntax information, see the Javadoc for java.util.regex.Pattern.

Value: Specify the value you want to compare with the Authentication Method value. If you select a static value for the Authentication Method value, select **Authentication Method** and **Current**. If you select **Current** for the Authentication Method value, select **Authentication Method**, then select the name of a method.

Other value types are possible if you selected **Current** for the Authentication Method value. Your policy requirements determine whether they are useful.

Result on Condition Error: Specify what the condition returns when the comparison of the two values returns an error rather than the results of the comparison. Select either **False** or **True**. If you do not want the action applied when an error occurs, select **False**. If you want the action applied when an error occurs, select **True**.

Authentication Type Condition

The Authentication Type condition allows you to assign a role based on the authentication types used to authenticate the current user. The [Current] selection represents the current set of authentication types used to authenticate the user. The other selections represent specific authentication types that can be used to compare with [Current]. The Authentication Type condition returns true if the selected Authentication Type is contained in the set of Authentication Types for [Current]. For example, if the current user was required to satisfy the Authentication Types of Basic and SmartCard, then a selected Authentication Type of either Basic or SmartCard would match.

Authentication Type: To compare the type that the user used with a static value, select **Current**. To compare a static value with what the user used, select a type from the list.

Comparison: Specify how the type is compared to the data in the **Value** field. Select either a string comparison or a regular expression:

- **Comparison: String:** Specifies that you want the values compared as strings and how you want the string values compared. Select one of the following:
  - **Equals:** Indicates that the values must match, letter for letter.
  - **Starts with:** Indicates that the Authentication Type value must begin with the letters specified in the **Value** field.
  - **Ends with:** Indicates that the Authentication Type value must end with the letters specified in the **Value** field.
  - **Contains Substring:** Indicates that the Authentication Type value must contain the letters, in the same sequence, as specified in the **Value** field.
- **Comparison: Regular Expression: Matches:** Specifies that you want the values compared as regular expressions.

Mode: Select the mode appropriate for the comparison type:

- **Comparison: String:** Specify whether case is important by selecting **Case Sensitive** or **Case Insensitive**.
- **Comparison: Regular Expression: Matches:** Select one or more of the following:

  Canonical Equivalence
  Case Insensitive
  Comments
  Dot All
  Multi-Line
  Unicode
  Unix Lines

  For regular expression syntax information, see the Javadoc for java.util.regex.Pattern.

Value: Specify the value you want to compare with the Authentication Type value. If you select a static value for the Authentication Type value, select **Authentication Type** and **Current**. If you select **Current** for the Authentication Type value, select **Authentication Type**, then select a type.

Other value types are possible if you selected **Current** for the Authentication Type value. Your policy requirements determine whether they are useful.

Result on Condition Error: Specify what the condition returns when the comparison of the two values returns an error rather than the results of the comparison. Select either **False** or **True**. If you do not want the action applied when an error occurs, select **False**. If you want the action applied when an error occurs, select **True**.

Credential Profile Condition

The Credential Profile condition allows you to assign a role based on the credentials the user entered when authenticating to the system. Only values used at authentication time are available for this comparison.

To set up the matching for this condition, fill in the following fields:

Credential Profile: Specify the type of credential your users are using for authentication. If you have created a custom contract that uses a credential other than the ones listed below, do not use the Credential Profile as a Role condition.

- **LDAP Credentials:** If you prompt the user for a username, select this option, then select **LDAP User Name** (the cn of the user) or **LDAP User DN** (the fully distinguished name of the user), or **LDAP Password**.

  The default contracts assign the cn attribute to the Credential Profile. If your user store is an Active Directory server, the SAMAccountName attribute is used for the username and stored in the cn field of the LDAP Credential Profile.
- **X509 Credentials:** If you prompt the user for a certificate, select this option, then select one of the following:
  - **X509 Public Certificate Subject:** Retrieves the subject field from the certificate, which can match the DN of the user, depending upon who issued the certificate.
  - **X509 Public Certificate Issuer:** Retrieves the issuer field from the certificate, which is the name of the certificate authority (CA) that issued the certificate.
  - **X509 Public Certificate:** Retrieves the entire certificate, Base64 encoded.
  - **X509 Serial Number:** Retrieves the serial number of the certificate.
- **SAML Credential:** If your users authenticate with a SAML assertion, select this option.

Comparison: Select one of the following types:

- **Comparison: String:** Specifies that you want the values compared as strings and indicates how you want the string values compared. Select one of the following:
  - **Equals:** Indicates that the values must match, letter for letter.
  - **Starts with:** Indicates that the Credential Profile value must begin with the letters specified in the **Value** field.
  - **Ends with:** Indicates that the Credential Profile value must end with the letters specified in the **Value** field.
  - **Contains Substring:** Indicates that the Credential Profile value must contain the letters, in the same sequence, as specified in the **Value** field.
- **Comparison: Regular Expression: Matches:** Specifies that you want the values compared as regular expressions.

Mode: Select the mode appropriate for the comparison type:

- **Comparison: String:** Specify whether case is important by selecting **Case Sensitive** or **Case Insensitive**.
- **Comparison: Regular Expression: Matches:** Select one or more of the following:

  Canonical Equivalence
  Case Insensitive
  Comments
  Dot All

Multi-Line
Unicode
Unix Lines

For regular expression syntax information, see the Javadoc for java.util.regex.Pattern.

Value: Specify the second value for the comparison. Select one of the following data types:

- **LDAP Attribute:** If you have an LDAP attribute that corresponds to the Credential Profile you have specified, select this option and the attribute.
- **Liberty User Profile:** If you have a Liberty User Profile attribute that corresponds to the Credential Profile you have specified, select this option and the attribute.
- **Data Entry Field:** Specify the string you want matched. Be aware of the following requirements:
  - If you selected **LDAP User DN** as the credential, you need to specify the DN of the user in the **Value** text box. If the comparison type is set to **Contains Substring**, you can match a group of users by specifying a common object that is part of their DNs, for example `ou=sales`.
  - If you selected **X509 Public Certificate Subject** as the credential, you need to specify all elements of the Subject Name of the certificate in the **Value** text box. Separate the elements with a comma and a space, for example, `o=novell, ou=sales`. If the comparison type is set to **Contains Substring**, you can match a group of certificates by specifying a name that is part of the Subject Name, for example `ou=sales`.

Other values are possible. Your policy requirements determine whether they are useful.

Result on Condition Error: Specify what the condition returns when the comparison of the two values returns an error rather than the results of the comparison. Select either **False** or **True**. If you do not want the action applied when an error occurs, select **False**. If you want the action applied when an error occurs, select **True**.

LDAP Group Condition

The LDAP Group condition allows you to assign a role based on whether the authenticating user is a member of a group. The value, an LDAP DN, must be a fully distinguished name of a group.

LDAP Group: Select **[Current]**.

Comparison: Specify how you want the values compared. Select one of the following:

- **LDAP Group: Is Member of:** Specifies that you want the condition to determine whether the user is member of a specified group.
- **Comparison: Regular Expression: Matches:** Specifies that you want the values compared as regular expressions.

Mode: If you selected **Regular Expression: Matches** as the comparison type, select one or more of the following:

Canonical Equivalence
Case Insensitive
Comments
Dot All
Multi-Line
Unicode
Unix Lines

For regular expression syntax information, see the Javadoc for java.util.regex.Pattern.

Value: Specify the second value for the comparison. If you select **LDAP Group** > **Name of Identity Server Configuration** > **User Store Name**, you can browse to the name of the LDAP group.

If you have more than 250 groups in your tree, you are prompted to enter an LDAP query string. In the text box, you need to add only the <strFilter> value for the query. For example:

<strFilter> Value	Description
admin*	Returns all groups that begin with admin, such as adminPR, adminBG, and adminWTH.
*test	Returns all groups that end with test, such as doctest, softtest, and securtest.
low	Returns all groups that have “low” in the name, such as low, yellow, and clowns.

For more information about the <strFilter> parameter, see RFC 2254 “LDAP Search Filter.”

If you select **Data Entry Field** as the value, you can specify the DN of the group in the text field. For example:

```
cn=managers,cn=users,dc=bcf2,dc=provo,dc=novell,dc=com
```

```
cn=manager,o=novell
```

Other values are possible. Your policy requirements determine whether they are useful.

Result on Condition Error: Specify what the condition returns when the comparison of the two values returns an error rather than the results of the comparison. Select either **False** or **True**. If you do not want the action applied when an error occurs, select **False**. If you want the action applied when an error occurs, select **True**.

LDAP OU Condition

The LDAP OU condition allows you to assign a role based on a comparison of the DN of an OU against the DN of the authenticated user. If the user’s DN contains the OU, the condition matches.

LDAP OU: Select **[Current]**.

Comparison: Specify how you want the values compared. Select one of the following:

- **Contains:** Specifies that you want the condition to determine whether the user is contained by a specified organizational unit.
- **Comparison: Regular Expression: Matches:** Specifies that you want the values compared as regular expressions.

Mode: Select the mode appropriate for the comparison type.

- **Contains:** Select whether the user must be contained in the specified OU (**One Level)** or whether the user can be contained in the specified OU or a child container (**Subtree)**.
- **Comparison: Regular Expression: Matches:** Select one or more of the following:

  Canonical Equivalence
  Case Insensitive
  Comments
  Dot All
  Multi-Line

Unicode

Unix Lines

For regular expression syntax information, see the Javadoc for java.util.regex.Pattern.

Value: Specify the second value for the comparison. If you select **LDAP OU** > **Name of Identity Server Configuration** > **User Store Name**, you can browse to the name of the OU.

If you have more than 250 OUs defined in your tree, you are prompted to enter an LDAP query string. In the text box, you need to add only the <strFilter> value for the query. For example:

<strFilter> Value	Description
admin*	Returns all OUs that begin with admin, such as adminPR, adminBG, and adminWTH.
*test	Returns all OUs that end with test, such as doctest, softtest, and securtest.
low	Returns all OUs that have “low” in the name, such as low, yellow, and clowns.

For more information about the <strFilter> parameter, see RFC 2254 “LDAP Search Filter.”

If you select **Data Entry Field,** you can specify the DN of the OU in the text field. For example:

```
cn=users,dc=bcf2,dc=provo,dc=novell,dc=com
```

```
ou=users,o=novell
```

If you have defined a Liberty User Profile or an LDAP attribute for the OU you want to match, select this option, then select your attribute.

Result on Condition Error: Specify what the condition returns when the comparison of the two values returns an error rather than the results of the comparison. Select either **False** or **True**. If you do not want the action applied when an error occurs, select **False**. If you want the action applied when an error occurs, select **True**.

LDAP Attribute Condition

The LDAP Attribute condition allows you to assign a role based on a value in an LDAP attribute defined for the inetOrgPerson class or any other LDAP attribute you have added. You can have the user’s attribute value retrieved from your LDAP directory and compared to a value of the following type:

- Roles from an identity provider
- Authenticating IDP or user store
- Authentication contract, method, or type
- Credential profile
- LDAP attribute, OU, or group
- Liberty User Profile attribute
- Static value in a data entry field

To set up the matching for this condition, fill in the following fields:

LDAP Attribute: Specify the LDAP attribute you want to use in the comparison. Select from the listed LDAP attributes. To add an attribute that isn't in the list, click **New LDAP Attribute**, then specify the name of the attribute.

Comparison: Specify how you want the values compared. All data types are available. Select one that matches the value type of your attribute.

Mode: Select the mode, if available, that matches the comparison type. For example, if you select to compare the values as strings, you can select either a **Case Sensitive** mode or a **Case Insensitive** mode.

Value: Specify the second value for the comparison. All data types are available. For example, you can select to compare the value of one LDAP attribute to the value of another LDAP attribute. Only you can determine if such a comparison is meaningful.

Result on Condition Error: Specify what the condition returns when the comparison of the two values returns an error rather than the results of the comparison. Select either **False** or **True**. If you do not want the action applied when an error occurs, select **False**. If you want the action applied when an error occurs, select **True**.

Liberty User Profile Condition

The Liberty User Profile condition allows you to assign a role based on a value in a Liberty User Profile attribute. The Liberty attributes must be enabled before you can use them in policies (click **Identity Servers** > **Edit** > **Liberty** > **Web Service Provider**, then enable one or more of the following: **Employee Profile** or **Personal Profile**).

These attributes can be mapped to LDAP attributes (click **Identity Servers** > **Edit** > **Liberty** > **LDAP Attribute Mapping**). When mapped, the actual value comes from your user store. If you are using multiple user stores with different LDAP schemas, mapping similar attributes to the same Liberty User Profile attribute allows you to create one policy with the Liberty User Profile attribute rather than multiple policies for each LDAP attribute.

The selected attribute is compared to a value of the following type:

- Roles from an identity provider
- Authenticating IDP or user store
- Authentication contract, method, or type
- Credential profile
- LDAP attribute, OU, or group
- Liberty User Profile attribute
- Static value in a data entry field

To set up the matching for this condition, fill in the following fields:

Liberty User Profile: Select the Liberty User Profile attribute. These attributes are organized into three main groups: Custom Profile, Corporate Employment Identity, and Entire Personal Identity. By default, the Common Last Name attribute for Liberty User Profile is mapped to the sn attribute for LDAP. To select this attribute for comparison, click **Entire Personal Identity** > **Entire Common Name** > **Common Analyzed Name** > **Common Last Name**.

Comparison: Select the comparison type that matches the data type of the selected attribute and the value.

Mode: Select the mode, if available, that matches the data type. For example, if you select to compare the values as strings, you can select either a **Case Sensitive** mode or a **Case Insensitive** mode.

Value: Select one of the values that is available from the current request or select **Data Entry Field** to enter a static value. The static value that you can enter depends on the comparison type you selected.

Result on Condition Error: Specify what the condition returns when the comparison of the two values returns an error rather than the results of the comparison. Select either **False** or **True**. If you do not want the action applied when an error occurs, select **False**. If you want the action applied when an error occurs, select **True**.

Roles from Identity Provider Condition

The Roles from Identity Provider condition allows you to assign a role based on a role assigned by another identity provider (Liberty, SAML 2.0, WS Federation). You configure the condition to match the role sent by the identity provider, then set the action to assign a new role.

This condition uses the mapped attribute All Roles. All roles that are assigned to the user can be mapped to attributes and assigned to a trusted identity provider. For information about enabling All Roles, see "Selecting Attributes for a Trusted Provider" on page 155.

For an example of how to use Roles from Identity Provider to create a Role policy, see "Mapping Roles between Trusted Providers" on page 640. For an example that explains all the configuration procedures required for sharing roles, see "Sharing Roles" on page 385.

To configure a Roles from Identity Provider condition, fill in the following fields:

Roles from Identity Provider: If you have configured your system for multiple identity providers, select the identity provider. If you have only one, it is selected.

Comparison: Select one of the following types:

- **Comparison: String:** Specifies that you want the values compared as strings, and how you want the string values compared. Select one of the following:
  - **Equals:** Indicates that the values must match, letter for letter.
  - **Starts with:** Indicates that the Roles from Identity Provider value must begin with the letters specified in the **Value** field.
  - **Ends with:** Indicates that the Roles from Identity Provider value must end with the letters specified in the **Value** field.
  - **Contains Substring:** Indicates that the Roles from Identity Provider value must contain the letters, in the same sequence, as specified in the **Value** field.
- **Comparison: Regular Expression: Matches:** Specifies that you want the values compared as regular expressions.

Mode: Select the mode appropriate for the comparison type:

- **Comparison: String:** Specify whether case is important by selecting **Case Sensitive** or **Case Insensitive**.
- **Comparison: Regular Expression: Matches:** Select one or more of the following:

  Canonical Equivalence
  Case Insensitive
  Comments
  Dot All
  Multi-Line
  Unicode

Unix Lines

For regular expression syntax information, see the Javadoc for java.util.regex.Pattern.

Value: Select **Data Entry Field**, then specify the name of an identity provider role. Other value types are possible. Your policy requirements determine whether they are useful

Result on Condition Error: Specify what the condition returns when the comparison of the two values returns an error rather than the results of the comparison. Select either **False** or **True**. If you do not want the action applied when an error occurs, select **False**. If you want the action applied when an error occurs, select **True**.

User Store Condition

The User Store condition allows you to assign a role based on the user store that was used to authenticate the current user. The [Current] selection represents the user store from which the user was authenticated. The other selections represent all of the configured user stores that can be used to compare with [Current].

For example, if the configured user stores are eDir1 and AD1 and the current user is authenticated from eDir1, then a selected user store of eDir1 would match and a selected user store of AD1 would not match.

User Store: To compare the user store that the user used for authentication with a static value, select **Current**. To compare a static value with what the user used, select a user store from the list.

If you have created more than one Identity Server configuration, select the configuration, then select the user store. The name of the user store is displayed.

Comparison: Specify how the user store is compared to the data in the **Value** field. Select either a string comparison or a regular expression:

- **Comparison: String:** Specifies that you want the values compared as strings and how you want the string values compared. Select one of the following:
  - **Equals:** Indicates that the values must match, letter for letter.
  - **Starts with:** Indicates that the User Store value must begin with the letters specified in the **Value** field.
  - **Ends with:** Indicates that the User Store value must end with the letters specified in the **Value** field.
  - **Contains Substring:** Indicates that the User Store value must contain the letters, in the same sequence, as specified in the **Value** field.
- **Comparison: Regular Expression: Matches:** Specifies that you want the values compared as regular expressions.

Value: Specify the value you want to compare with the User Store value. If you select a static value for the User Store value, select **User Store** and **Current**. If you select **Current** for the User Store value, select **User Store**, then select the name of a user store.

If you have created more than one Identity Server configuration, select the configuration, then select the user store. The name of the user store is displayed.

Other value types are possible if you selected **Current** for the User Store value. Your policy requirements determine whether they are useful.

Result on Condition Error: Specify what the condition returns when the comparison of the two values returns an error rather than the results of the comparison. Select either **False** or **True**. If you do not want the action applied when an error occurs, select **False**. If you want the action applied when an error occurs, select **True**.

Virtual Attribute Condition

The Virtual Attribute condition allows you to assign a role based on a value in an Virtual attribute. You can have the user's attribute value retrieved from an external source and compared to a value of the following type:

- Roles from an identity provider
- Authenticating IDP or user store
- Authentication contract, method, or type
- Credential profile
- LDAP attribute, OU, or group
- Liberty User Profile attribute
- Static value in a data entry field
- Virtual Attribute

To set up the matching for this condition, fill in the following fields:

Virtual Attribute: Specify the virtual attribute you want to use in the comparison. Select a virtual attribute from the list.'

Comparison: Specify how you want the values compared. All data types are available. Select one that matches the value type of your virtual attribute.

Mode: Select the mode, if available, that matches the comparison type. For example, if you select to compare the values as strings, you can select either a **Case Sensitive** mode or a **Case Insensitive** mode.

Value: Specify the second value for the comparison. All data types are available. For example, you can select to compare the value of one virtual attribute to the value of another virtual attribute. Only you can determine if such a comparison is meaningful.

Result on Condition Error: Specify what the condition returns when the comparison of the two values returns an error rather than the results of the comparison. Select either **False** or **True**. If you do not want the action applied when an error occurs, select **False**. If you want the action applied when an error occurs, select **True**.

Condition Extension

If you have loaded and configured a role condition extension, this option specifies a condition that is evaluated by an outside source. See the documentation that came with the extension for information about what is evaluated.

Data Extension

If you have loaded and configured a role data extension, this option specifies the value that the extension retrieves. You can then select to compare this value with an LDAP attribute, a Liberty User Profile attribute, a Data Entry Field, or another Data Extension. For more information, see the documentation that came with the extension.

Using Multiple Conditions

The **Condition structure** field controls how conditions within a condition group interact with each other and how condition groups interact with each other. Select one of the following:

- “AND Conditions, OR groups” on page 624
- “OR Conditions, AND groups” on page 624

The following sections explain how to configure the condition groups and conditions to interact with each other:

- “Using the Not Options” on page 625
- “Adding Multiple Conditions” on page 625
- “Adding New Condition Groups” on page 625
- “Disabling Conditions and Condition Groups” on page 625

AND Conditions, OR groups

If the conditions are ANDed, the user must meet all the conditions in a condition group to match the profile. If the condition groups are ORed, the user must meet all of the conditions of one group to match the profile. This option allows you to set up two or more profiles into which a user could fit and be considered a match. For example, suppose you create the following Permit rule.

The first condition group contains the following conditions:

1. The user’s department must be Engineering.
2. The request must come on a weekday.

The second condition group contains the following conditions:

1. The user’s department must be Information Services and Technology (IS&T).
2. The request must come on a weekend.

With this rule, the engineers who match the first condition group have access to the resource during the week, and the IS&T users who match the second condition group have access to the resource on the weekend.

OR Conditions, AND groups

If the conditions are ORed, the user must meet at least one condition in the condition group to match the profile. If the conditions groups are ANDed, the user must meet at least one condition in each condition group to match the profile. For example, suppose you created the following Permit rule:

The first condition group contains the following conditions:

1. The user’s department is Engineering.
2. The user’s department is Sales.

The second condition group contains the following conditions:

1. The user has been assigned the Party Planning role.
2. The user has been assigned the Vice President role.

With this rule, the Vice Presidents of both the Engineering and Sales departments can access the resource, and the users from the Engineering and Sales department who have been assigned to the Party Planning role can access the resource.

Using the Not Options

At the top of each condition group, there is an option that allows you to control whether the user must match the conditions to match the profile or whether the user matches the profile if the user does not' match any of the conditions. Depending upon your selection for the Condition structure, you can select from the following:

- If/If Not
- Or/Or Not
- And/And Not

Conditions also have similar Not options, so that a user can match a condition by not matching the specified value.

Adding Multiple Conditions

To add another condition to a condition group, click **New**, then select a condition. To copy an existing condition, click the **Copy Condition** icon. New conditions are always added to the end of the condition group. Use the **Move** buttons to order the conditions in the condition group.

Adding New Condition Groups

To add another condition group to the rule, click **Append New Group**. To copy the existing condition group, click the **Copy Group** icon. New condition groups are always added to the end to the Conditions section. Use the **Move** buttons to order the condition groups.

Disabling Conditions and Condition Groups

Condition groups and conditions within them can be disabled by clicking the Enabled check mark, which changes the icon to the **Disabled** icon.

You usually disable a condition or condition group when testing a new rule, and if you decide the condition or condition group is not needed, you can then use the **Delete** button to delete the condition or condition group from the rule. Use the **Move** buttons by the **Delete** button to move a condition up or down within its group. Condition groups also have **Move** buttons.

Selecting an Action

The policy action specifies the role to which the user is assigned. Roles are activated at the time the role policy is evaluated. Select one of the following actions:

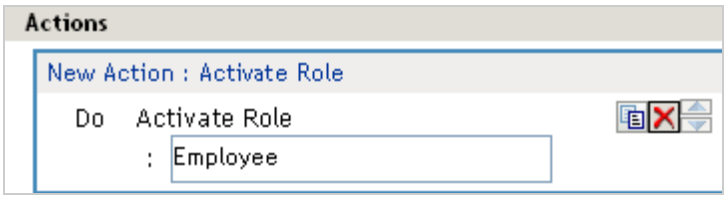
- "Activate Role" on page 625
- "Activate Selected Role" on page 626

Activate Role

Select **Activate Role** when you want to specify a name for the role. If you are creating a role that needs to injected into an HTTP header, use the same capitalization format as the Web server expects. For example, if the Web server expects an Employee role with an initial capital, name your role Employee.

Figure 6-8 show how to assign the role of Employee to a policy.

Figure 6-8 Assigning a Role



To use the same conditions to activate multiple roles, select **Activate Role** for each role you want to specify.

Activate Selected Role

Select **Activate Selected Role** when you want to obtain the role value from an external source. Select one of the following:

- **LDAP Attribute:** If you have an LDAP attribute that is a role, select the attribute from the list. If the attribute is not in the list, select **New LDAP Attribute** to add it to the list.
- **LDAP Group:** Activates a role based on an LDAP Group attribute. Select either [Current] or browse to the DN of the group by selecting the Identity Server and User Store. The value for this option is the DN of the group. If you select [Current], the value can be a list of the groups the user belongs to. The [Current] value makes the DN of each group in the attribute into a role.

  If you select to browse to the DN of the group and you have more than 250 groups in your tree, you are prompted to enter an LDAP query string. In the text box, you need to add only the <strFilter> value for the query. For example:

<strFilter> Value	Description
admin*	Returns all groups that begin with admin, such as adminPR, adminBG, and adminWTH.
*test	Returns all groups that end with test, such as doctest, softtest, and securtest.
low	Returns all groups that have “low” in the name, such as low, yellow, and clowns.

  For more information about the <strFilter> parameter, see RFC 2254 “LDAP Search Filter.”

  This action does not query all the static and dynamic groups on the LDAP server to see if the user belongs to them, but uses the user’s group membership attribute to create the list. If you want to use this longer query, you need to create a policy extension. For a sample extension that does this, see Access Manager SDK Sample Code.

- **LDAP OU:** Activates a role based on the Organizational Unit in the user’s DN. Select either [Current] or browse to the DN of the OU by selecting the Identity Server and User Store. The value for this option is the DN of the OU.

  If you select to browse to the DN of the OU and you have more than 250 OUs defined in your tree, you are prompted to enter an LDAP query string. In the text box, you need to add only the <strFilter> value for the query. For example:

<strFilter> Value	Description
admin*	Returns all OUs that begin with admin, such as adminPR, adminBG, and adminWTH.
*test	Returns all OUs that end with test, such as doctest, softtest, and securtest.
low	Returns all OUs that have “low” in the name, such as low, yellow, and clowns.

For more information about the <strFilter> parameter, see RFC 2254 “LDAP Search Filter.”

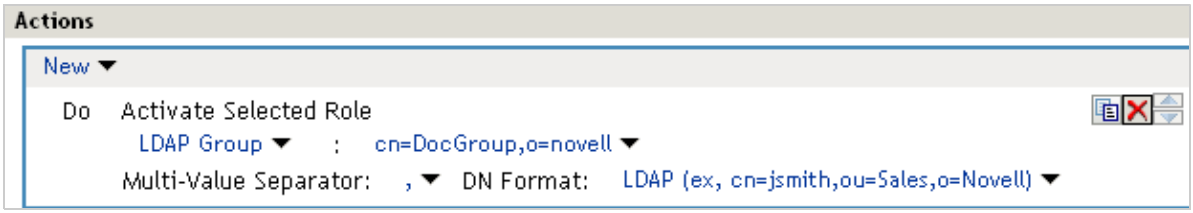
- **Liberty User Profile:** If you have a Liberty attribute that is a role, select the attribute from the list.
- **Data Extension:** If you have created a data extension that calculates a set of roles, select the extension. For information about creating such an extension, see Access Manager SDK Sample Code.

If the source contains multiple values, select the format that is used to separate the values.

If the value is a distinguished name, select the format of the DN.

Figure 6-9 shows how to assign an LDAP Group, cn=DocGroup,o=novell, as a role.

Figure 6-9 *Activating a Role from an External Source*



To use the same conditions to activate multiple roles from different sources, select **Activate Selected Role** for each role you want to activate.

Example Role Policies

The following examples describe how to create a general Employee role, a restrictive Manager role, and a role from a contract with ORed credentials. These roles can be used by the Access Gateway in Identity Injection policies and by the Access Gateway in Authorization policies.

- “Creating an Employee Role” on page 628
- “Creating a Manager Role” on page 628
- “Creating a Rule for a Contract with ORed Credentials” on page 629

Creating an Employee Role

The following role policy creates an Employee role. Because the role does not include conditions, all authenticated users are assigned to this role when they log in. This role can then be used to grant access to resources to all users in your user stores.

1. In the Administration Console, click **Devices > Identity Servers** > *Edit* > **Roles > Manage Policies.**
2. On the Policies page, click **New**.
3. Select a policy type of **Identity Server: Roles** and specify a display name, such as Employee.
4. Click **OK**.
5. On the Edit Policy page, specify a description in the **Description** field.

   It is important to use this field to keep track of your roles and policies. The policy feature is powerful, and your setup can be as large and complex as you want it to be, with a potentially unlimited number of conditions and choices. This description is useful to help keep track of various role and policy configurations.
6. Ensure that the **Condition Group 1** section has no conditions, so that all users who authenticate match the condition.
7. In the **Actions** section, click **New > Activate Role**.
8. In the **Activate Role** box, type `Employee`, then click **OK**.

   If this role needs to match the name of a role required by a Java or Web application, ensure that the case of the name matches the application's name.
9. On the Rule List page, click **OK**.
10. On the Policies page, click **Apply Changes**, then click **Close**.
11. On the Role Policy page, select the Employee role, then click **Enable.**
12. Click **OK**, then update the Identity Server.

    The Identity Server configuration must be updated after you enable a role.
13. To create a Manager role, continue with "Creating a Manager Role" on page 628.

Creating a Manager Role

Because the Manager role is restrictive, role policy conditions must be specified. The Manager role is assigned only to the users who meet the conditions.

1. In the Administration Console, click **Devices** > **Identity Servers** > *Edit* > **Roles > Manage Policies.**
2. On the Policies page, click **New**.
3. Select a policy type of **Identity Server: Roles** and specify a display name (for this example, Manager.)
4. Click **OK**.
5. In the **Conditions** section, click **New** > **Liberty User Profile**.
6. In **Condition Group 1**, select the conditions the user must meet:

   Liberty User Profile: Select **Entire Personal Identity > Entire Common Name > Common Analyzed Name > Common Last Name.**

   If these options are not available, you haven't enabled the Liberty attributes. Click **Identity Servers** > **Edit** > **Liberty** > **Web Service Provider**, then enable one or more of the following: **Employee Profile** or **Personal Profile.**

Comparison: Select how you want the attribute values to be compared. For the Common Last Name attribute, select **String > Equals**.

Mode: Select **Case Insensitive**.

Value: Select **Data Entry Field** and type the person's name in the box (Smith, in this example). This sets up the condition that if the user has the name Smith, his or her role as Manager is activated at authentication.

Result on Condition Error: This sets up the results that are returned if an error occurs while evaluating the condition (for example, the LDAP server goes down). This rule is set up to grant the user the role of Manager if the condition evaluates to **True**. If an error occurs, you do not want random users assigned the role of Manager. Therefore, for this rule, you need to select **False**.

7 In the **Actions** section, click **Activate Role**.

8 In the **Activate Role** box, type `Manager`, then click **OK** twice.

9 On the Policies page, click **Apply Changes**.

10 Click **Close**, select the Manager role, then click **Enable.**

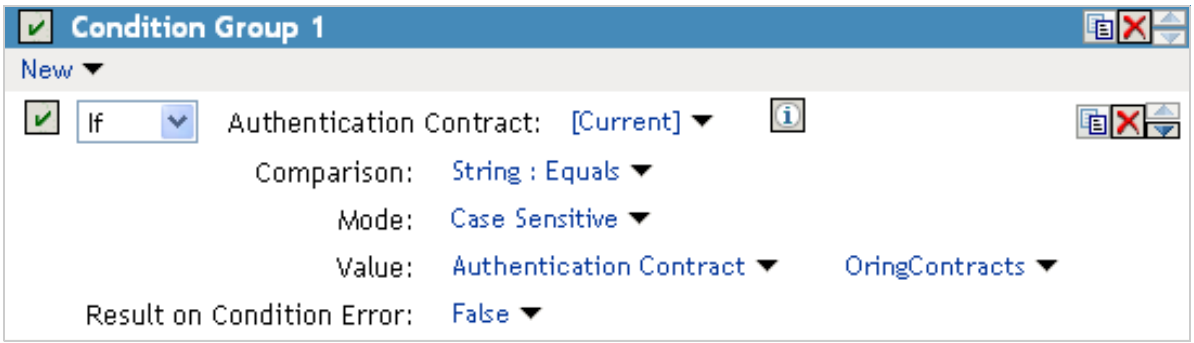
11 Click **OK**, then update the Identity Server.

Creating a Rule for a Contract with ORed Credentials

A contract with ORed credentials allows the user to decide which credentials to use for authenticating. If you are creating a role policy that grants the user the role regardless of which method was used for authentication, you can use such a contract just as you would any other contract in a condition. However, if you want to base the condition on the user using the contract with multiple credentials for authentication and on the user authenticating with a particular credential (password, token, or certificate), you need to create a rule with two conditions: one condition checks for the contract and the second condition checks for the authenticating credential.

If the contract with ORed credentials was named OringContracts, the first condition in the rule must look similar to the following:

Figure 6-10 *Checking for the Contract*



This condition verifies that the user used the OringContracts contract for authentication. The second condition needs to verify the type of credential that was used. To do this, you need to check for the existence of the credential in the Credential Profile. This condition must look similar to the following if you are verifying that the user used a certificate for the credential.

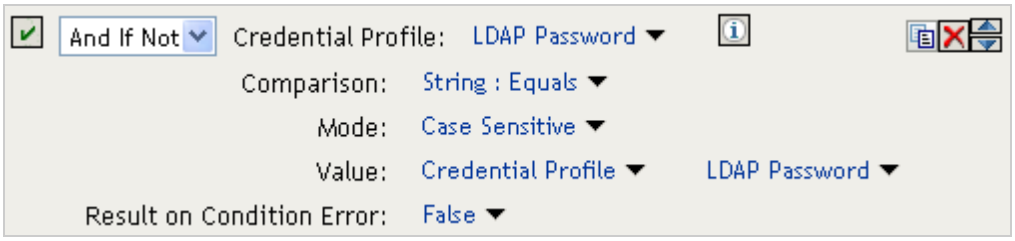
Figure 6-11 *Checking for the Credential*



The policy engine evaluates the above condition to true when the Credential Profile contains a value for the certificate. If the user used another method for authentication, the certificate field is empty, and the policy engine evaluates two null entries to false.

This type of condition works for the LDAP credentials and the X.509 credentials. It does not work for the Radius token, because the Credential Profile does not store the Radius token. You need to use “If Not” logic to verify that the user authenticated with a token. For example, if the OringContracts contract ORed the Radius token class with the Name/Password class, you would know that the user authenticated with a token when the password credential has no value. This type of condition must look similar to the following:

Figure 6-12 *Using “If Not” Logic*



If the Credential Profile contains a value for the password, this condition evaluates to false because of the “And If Not” logic. If the password value in the Credential Profile is empty, this condition evaluates to true, and you know that the user authenticated with a Radius token.

Creating Access Manager Appliance Roles in an Existing Role-Based Policy System

If you have already implemented a role-based administration policy for granting access to print, file, and LDAP resources, you can leverage your role definitions and use Access Manager Appliance policies to control access to Web resources. If your role definitions use the following types of LDAP features, you can create Access Manager Appliance Role policies that use them:

- Values found in LDAP attributes
- Location of the user objects in the directory tree
- Membership in groups or roles

The Access Manager Appliance Role policies that you create for these features can then be used to control access to protected Web resources. You can manually assign the roles by creating role policies with conditions or you can activate roles based on the values in the external source.

- “Activating Roles from External Sources” on page 631
- “Using Conditions to Assign Roles” on page 632

Activating Roles from External Sources

If you have an LDAP attribute, an LDAP group, an LDAP OU, or a Liberty attribute that you are currently using for role assignments, you can have Access Manager Appliance read its value and activate roles based on the values. This allows you to use the same roles for Access Manager Appliance access as you are using in other parts of your deployment.

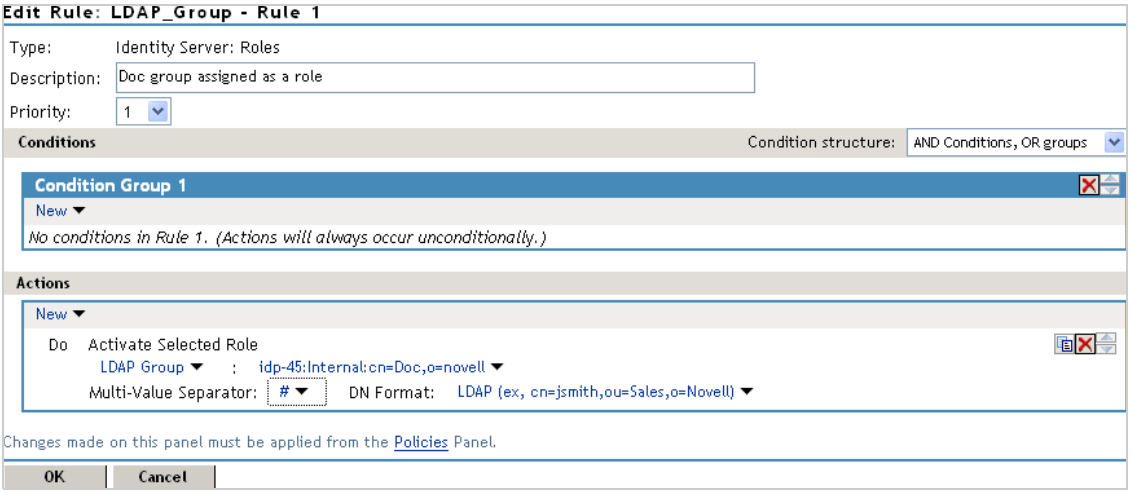
When you create this type of Role policy, you do not need to specify any conditions. The policy engine reads the attribute you specify, then assigns roles to users based on the value or values in the attribute. If the user has no value for the attribute, the user is assigned no roles. If the user has a value for the attribute, the user is assigned a role for each value in the attribute.

1. In the Administration Console, click **Policies** > **Policies**.
2. Select the policy container, then click **New** to create a new policy.
3. Specify a name for the Role policy, select **Identity Server: Roles** for the type, then click **OK**.
4. On the Rule page in the **Actions** section, click **New** > **Activate Selected Role**.
5. For this example, select **LDAP Group**.
6. To select the group you want to use for role assignments, click **Current** > **[Identity Server Name]** > **[User Store Name]** > **[Group Name]**.

   The distinguished name of this group is the Role name that is assigned to the user.
7. Select a **Multi-Value Separator** that is compatible with a distinguished name.

   A comma, which is the default separator, cannot be used because a comma is used to separate the components in a distinguished name. Select any other value, such as #.

   Your policy must look similar to the following:



8. Click **OK** twice, then click **Apply Changes**.
9. To enable the role so that it can be used in Authorization and Identity Injection policies, click **Devices** > **Identity Servers** > **Edit > Roles**.
10. Select the check box next to the name of the role, then click **Enable**.
11. Click **OK**.
12. Update the Identity Server.
13. (Optional) Verify the name used for the role and the user assigned to it:

    13a Enable logging by clicking **Devices** > **Identity Servers** > **Edit** > **Logging**, then set the following values:

File Logging: Select **Enabled**.

Echo To Console: Select this option to enable it.

Application: In the **Component File Logger Levels** section, set to **info**.

13b Click **OK**, then update the Identity Server.

13c Log in to the Identity Server by using the credentials of a user who belongs the LDAP group.

13d View the log file for the Identity Server by clicking **Auditing** > **General Logging**

13e Select the file (for Windows, select the `stdout.log` file; for Linux, select the `catalina.out` file), then click **Download**.

13f Look for two log entries (<amLogEntry>) similar to the following:

```
<amLogEntry> 2009-10-09T21:58:55Z INFO NIDS Application: AM#500199050:
AMDEVICEID#CA50FD51DB1EEE3E: AMAUTHID#213E610199A14CEAF27395A6B35F3162:
IDP RolesPep.evaluate(), policy trace:
   ~~RL~1~~~~Rule Count: 1~~Success(67)
   ~~RU~RuleID_1223587171711~LDAP_Group~DNF~~0:1~~Success(67)
   ~~PA~ActionID_1223588319336~~AddSelectedRoles~cn=Doc~~~Success(0)
   ~~PA~ActionID_1223588319336~~AddSelectedRoles~o=novell~~~Success(0)
   ~~PC~ActionID_1223588319336~~Document=(ou=xpemlPEP,ou=mastercdn,
ou=ContentPublisherContainer,ou=Partition,ou=PartitionsContainer,
ou=VCDN_Root,ou=accessManagerContainer,o=novell:romaContentCollection
XMLDoc),Policy=(LDAP_Group),Rule=(1::RuleID_1223587171711),Action=
(AddSelectedRole::ActionID_1223588319336)~~~~Success(0)
 </amLogEntry>

<amLogEntry> 2009-10-09T21:58:55Z INFO NIDS Application: AM#500105013:
AMDEVICEID#CA50FD51DB1EEE3E: AMAUTHID#213E610199A14CEAF27395A6B35F3162:
Authenticated user cn=jwilson,o=novell in User Store Internal with roles
"cn=Doc,o=novell","authenticated".
</amLogEntry>
```

The first <amLogEntry> entry indicates that the action in the LDAP_Group policy was successfully assigned.

The second entry gives the DN of the user and lists the roles assigned to the user: cn=Doc,o=novell and authenticated.

You can now use the cn=Doc,o=novell role when creating Authorization and Identity Injection policies, which control access to protected Web resources. Roles activated this way do not appear in the list of available roles. You need to use the **Data Entry Field** to manually type in the role name. For more information, see the following:

- Chapter , “Authorization Policies,” on page 642
- Chapter , “Identity Injection Policies,” on page 691

Using Conditions to Assign Roles

- “Creating a Role by Using an LDAP Attribute” on page 633
- “Creating a Role by Using the Location of the User Objects” on page 634
- “Creating a Role by Using a Group Membership Attribute” on page 637

Creating a Role by Using an LDAP Attribute

You can assign a user to a role by using a value found in any LDAP attribute in your directory. The following example uses the objectClass attribute because every object in an LDAP directory has an objectClass attribute that contains the object classes to which the object belongs. This attribute contains the name of the object class that was used to create the object as well as the names of the superior object classes of this class. All you need to know is the name of the object class you used to create your users in the LDAP directory. For example, the following instructions create a Role policy for users who were created with the User object class.

1. In the Administration Console, click **Policies** > **Policies**.
2. Select the policy container, then click **New**.
3. Specify a name for the Role policy, select **Identity Server: Roles** for the type, then click **OK**.
4. In **Condition Group 1**, click **New**, then select **LDAP Attribute**.
5. In **Condition Group 1**, select the conditions the user must meet:

   LDAP Attribute: Select the objectClass attribute. If you have not added this attribute, it won't appear in the list. Scroll to the bottom of the list, click **New LDAP Attribute**, specify objectClass for the name, then click **OK**.

   If you are using eDirectory™ for your LDAP directory, you need to specify standard LDAP names for the attributes. Access Manager Appliance does not support spaces or colons in attribute names.

   Comparison: Select how you want the attribute values to be compared. For the objectClass attribute, select **String** > **Contains Substring**.

   The objectClass attribute is a multi-valued attribute and, for most objects, contains multiple values. For example in eDirectory, users created with the User object class have User, organizationalPerson, person, ndsLoginProperties, and top as values in the objectClass attribute.

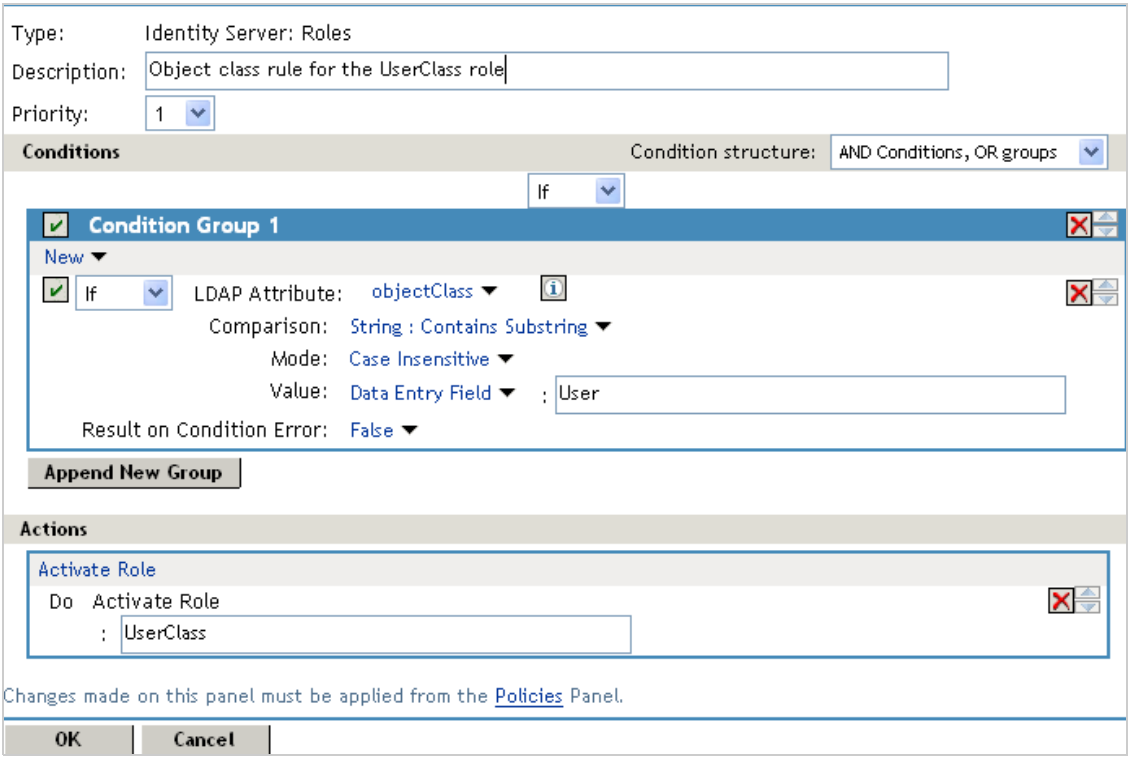
   Mode: Select **Case Insensitive**.

   Value: Select **Data Entry Field** and specify User as the value.

   Result on Condition Error: This sets up the results that are returned if an error occurs while evaluating the condition (for example, the LDAP server goes down). This rule is set up to grant the user the role of UserClass if the condition evaluates to **True**. If an error occurs, you do not want random users assigned the role of UserClass. Therefore, for this rule, you need to select **False**.
6. In the **Actions** section, click **Activate Role**.
7. In the **Activate Role** box, type `UserClass`, then click **OK.**

   The name you specify in the box is the role you want assigned to the users who match the condition.

   Your rule must look similar to the following:



8 Click **OK** twice, then click **Apply Changes**.

9 To enable the role so that it can be used in Authorization and Identity Injection policies, click **Identity Servers** > **Edit** > **Roles**.

10 Select the check box next to the name of the role, then click **Enable**.

11 Click **OK**.

12 Update the Identity Server.

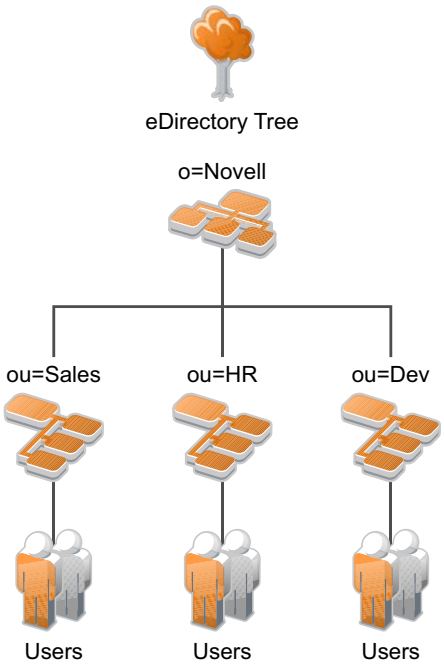
You can now use this role when creating Authorization and Identity Injection policies, which control access to protected Web resources. For more information, see the following:

- Chapter , “Authorization Policies,” on page 642
- Chapter , “Identity Injection Policies,” on page 691

Creating a Role by Using the Location of the User Objects

If you have created your users in specific containers in your LDAP tree, you can use these container objects to assign users to roles. For example, suppose your LDAP tree looks similar to the following tree.

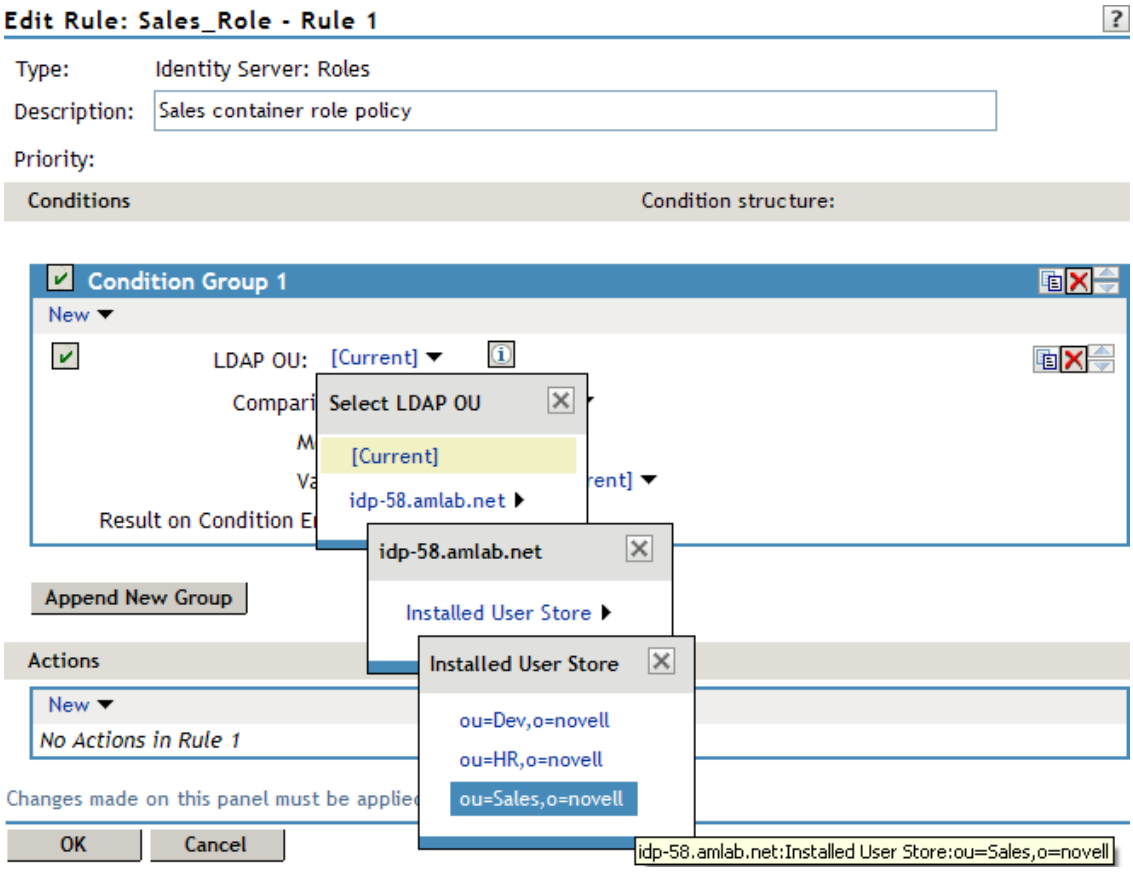
Figure 6-13 *Using an eDirectory Tree for Access Control*



Such a tree organization can be used to control access to resources. The following instructions explain how to create a Role policy for the users created under the Sales container.

1. In the Administration Console, click **Policies** > **Policies**.
2. Select the policy container, then click **New**.
3. Specify a name for the Role policy, select **Identity Server: Roles** for the type, then click **OK**.
4. In **Condition Group 1**, click **New**, and select **LDAP OU > [Identity Server Configuration] > [User Store] > [DN of the OU]**.

The following example illustrates how to make these selections:



Comparison: Select how you want the attribute values to be compared. For LDAP OU, select **Contains**.

Mode: Select **One Level** if all your users are created in ou=Sales. Select **Subtree** if your users are created in various containers under the ou=Sales container.

Value: Select **LDAP OU**, then select **[Current]**.

The DN of the authenticated user is compared with the value specified in LDAP OU. If the DN of the user contains the LDAP OU value, the user matches the condition. For example, if the DN of the user is cn=bsmith,ou=sales,o=novell and the LDAP OU value is ou=sales,o=novell, the user matches the condition. If you selected **Subtree** for the Mode, a user with the following DN also matches the condition: cn=djones,ou=provo,ou=sales,o=novell.

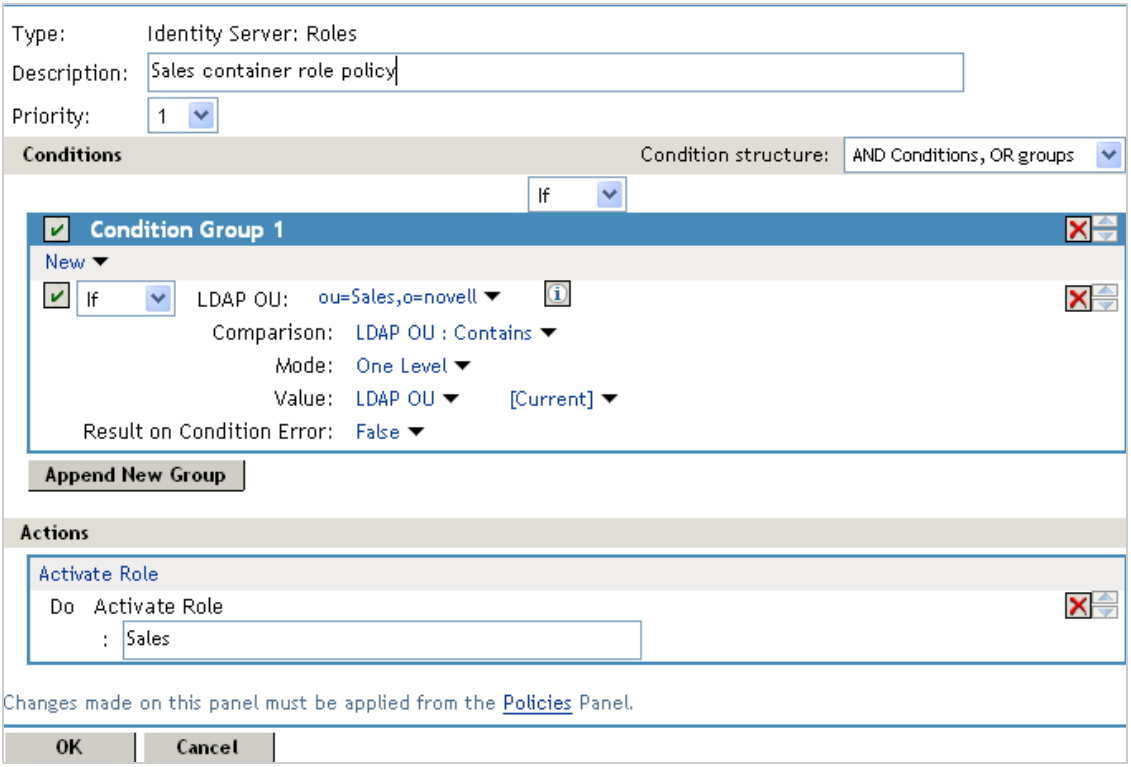
Result on Condition Error: This sets up the results that are returned if an error occurs while evaluating the condition (for example, the LDAP server goes down). This rule is set up to grant the user the role of Sales if the condition evaluates to **True**. If an error occurs, you do not want random users assigned the role of Sales. Therefore, for this rule, you need to select **False**.

5 In the **Actions** section, click **Activate Role**.

6 In the **Activate Role** box, type `Sales`, then click **OK**.

The name you specify in the box is the role you want assigned to the users who match the condition.

Your rule must look similar to the following:



7 Click **OK** twice, then click **Apply Changes**.

8 To enable the role so that it can be used in Authorization and Identity Injection policies, click **Devices > Identity Servers** > **Edit > Roles**.

9 Select the check box next to the name of the role, then click **Enable**.

10 Click **OK**.

11 Update the Identity Server.

You can now use this role when creating Authorization and Identity Injection policies, which control access to protected Web resources. For more information, see the following:

- Chapter , “Authorization Policies,” on page 642
- Chapter , “Identity Injection Policies,” on page 691

Creating a Role by Using a Group Membership Attribute

If you have created an LDAP group and assigned users to the group, you can use group membership to assign a role to the user. For example, you might have created a first-level managers group and made all your first-level managers members of this group. You can then have other groups for your upper-level managers. You can create a Role policy that assigns the user a role if the user is a member of a specific group. The Role policy can then be used in an Authorization or Identity Injection policy to protect a Web resource.

1 In the Administration Console, click **Policies** > **Policies**.

2 Select the policy container, then click **New**.

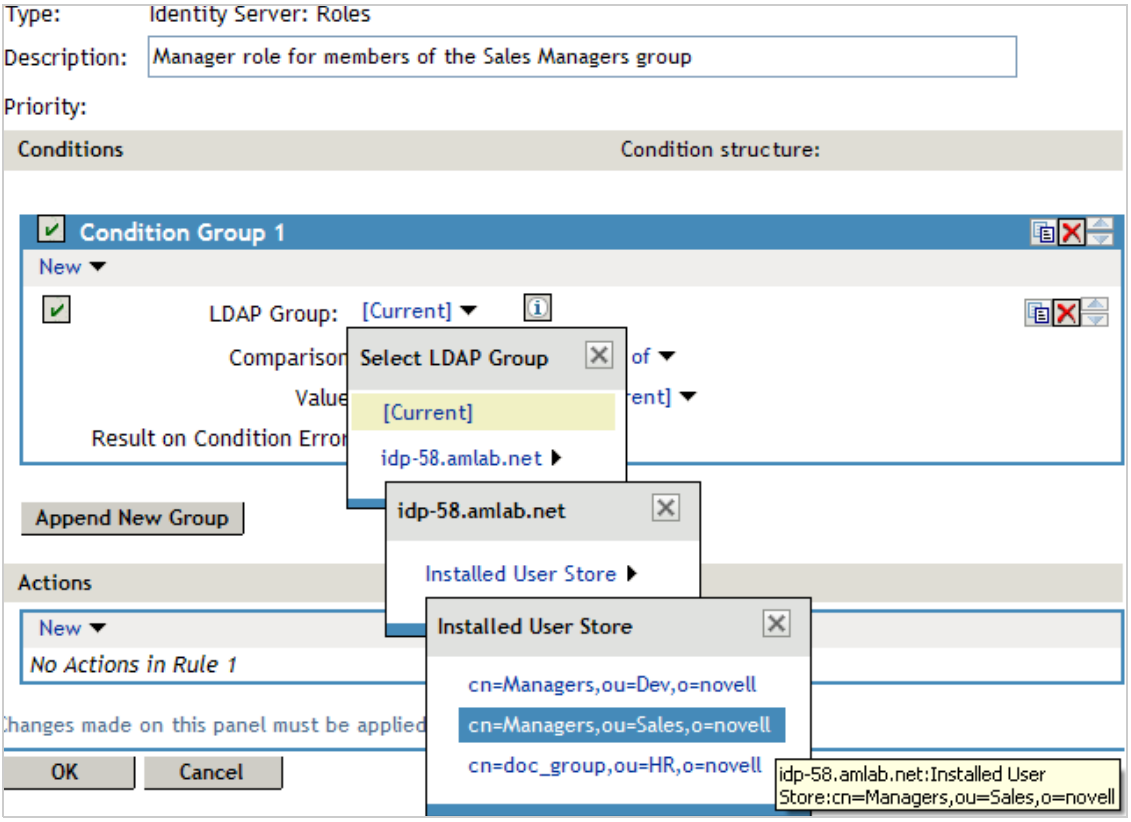
3 Specify a name for the Role policy, select **Identity Server: Roles** for the type, then click **OK**.

4 In **Condition Group 1**, click **New**, then select **LDAP Group**.

5 In **Condition Group 1**, select the conditions the user must meet:

LDAP Group: Select the Identity Server Configuration, the user store, then the Group.

The following figure illustrates this selection process.



Comparison: Select how you want the attribute values to be compared. For LDAP Group, select **Is Member of**.

Value: Select **LDAP Group**, then select **[Current]**.

The DN of the authenticated user is compared with the members of the LDAP Group. If the DN of the user matches one of the members, the user matches the condition.

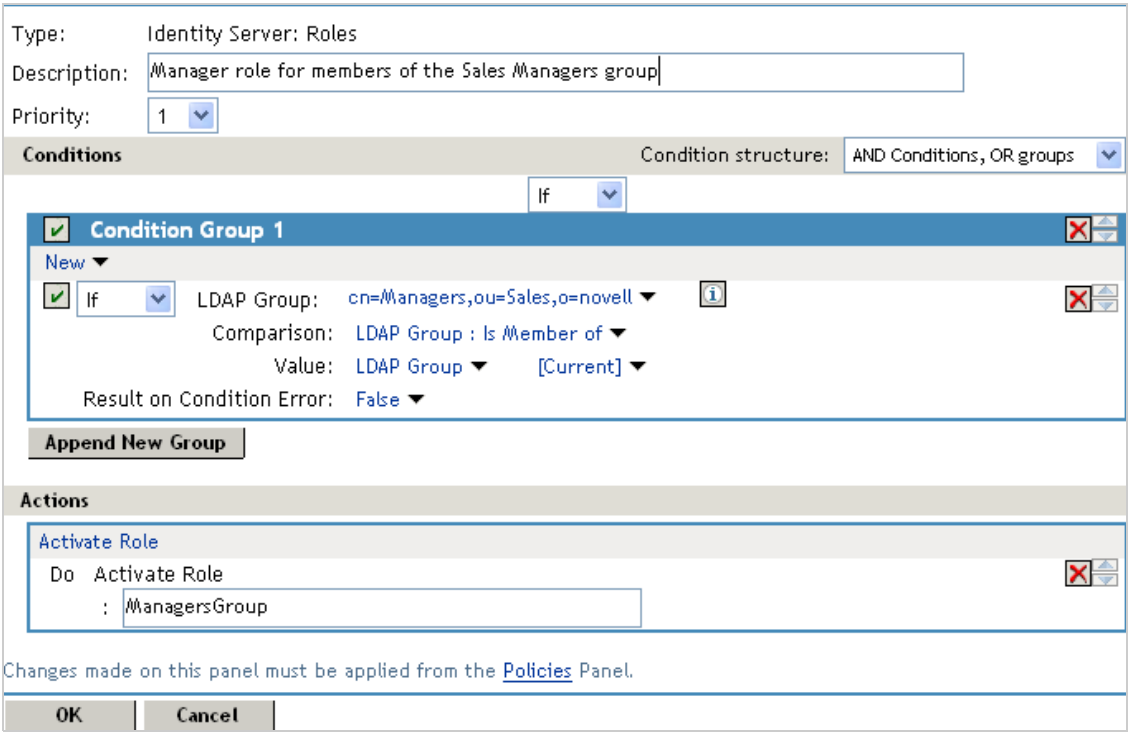
Result on Condition Error: This sets up the results that are returned if an error occurs while evaluating the condition (for example, the LDAP server goes down). This rule is set up to grant the user the role of ManagersGroup if the condition evaluates to **True**. If an error occurs, you do not want random users assigned the role of ManagersGroup. Therefore, for this rule, you need to select **False**.

6 In the **Actions** section, click **Activate Role**.

7 In the **Activate Role** box, type `ManagersGroup`, then click **OK**.

The name you enter in the box is the role you want assigned to the users who match the condition.

Your rule must look similar to the following:



8 Click **OK** twice, then click **Apply Changes**.

9 To enable the role so that it can be used in Authorization and Identity Injection policies, click **Devices > Identity Servers** > **Servers** > **Edit > Roles**.

10 Select the check box next to the name of the role, then click **Enable**.

11 Click **OK**.

12 Update the Identity Server.

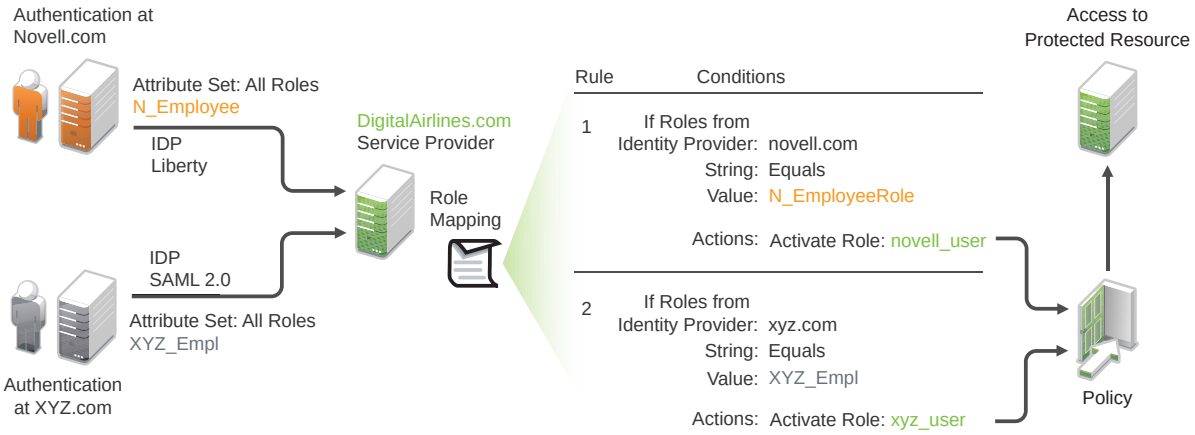
You can now use this role when creating Authorization and Identity Injection policies, which control access to protected Web resources. For more information, see:

- Chapter , “Authorization Policies,” on page 642
- Chapter , “Identity Injection Policies,” on page 691

Mapping Roles between Trusted Providers

The Identity Server can send roles in an authentication assertion. You can map these roles that are received from trusted providers to your own roles. Figure 6-14 illustrates this process.

Figure 6-14 *Role Mapping*



In this example, employees authenticate to identity providers novell.com (Liberty) or xyz.com (SAML 2.0). Each user is assigned to a role, such as N_EmployeeRole or XYZ_Empl. Attribute sets at each of the identity providers are configured to exchange the **All Roles** attribute with the trusted service provider, DigitalAirlines.com. DigitalAirlines.com consumes the authentication assertions, then maps the incoming roles to local roles. The mapped roles at DigitalAirlines.com can be used as evaluated conditions in authorization policies, which can provide access to resources intended for the authenticated employees.

- "Prerequisites" on page 640
- "Procedure" on page 641

Prerequisites

❒ Configure trust between trusted providers, using the Liberty or SAML 2.0 protocol.

You must be familiar with Chapter , "Configuring SAML 2.0," on page 418 and Chapter , "Configuring Liberty," on page 455.

❒ Configure local authentication.

You must create an external contract at the service provider that matches the contract of the identity provider. See Chapter , "Configuring Local Authentication," on page 271.

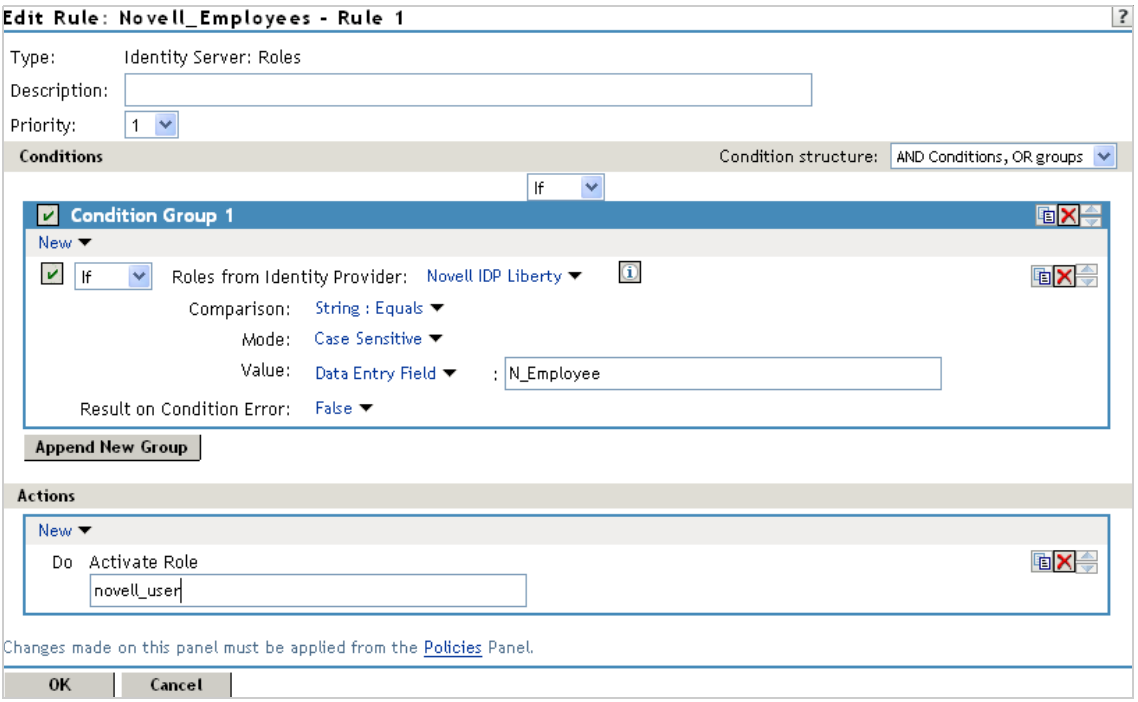
❒ Create an attribute set and select the local attribute **All Roles** in the set. This must be done at the identity provider and service provider.

This attribute set is used to pass roles from an identity provider to an external service provider in authentication assertions. See "Configuring Attribute Sets" on page 61.

Procedure

The following procedure describes how the service provider configures this type of role policy for novell.com, mapping the N_Employee role to an Access Manager Appliance role:

1. In the Administration Console, click **Policies** > **Policies**.
2. Click **New**, then specify a name for the Role policy.
3. Select **Identity Server: Roles** for the type, then click **OK**.
4. Configure the role policy as shown on the following page.



5. In the **Conditions** section, click **New > Roles from Identity Provider**.
6. Select the trusted identity provider in the drop-down menu.
7. For **Comparison**, select **String > Equals**.
8. Select **Value > Data Entry Field**.
9. Type the name of the role used by the trusted identity provider.
10. Under the **Actions** section, click **Activate Role**.
11. Type the name of the role you want to activate at the trusted service provider.
12. Click **OK**.
13. On the Policies page, click **Apply Changes**.
14. To enable the role so that it can be used in Authorization and Identity Injection policies, click **Identity Servers** > **Servers** > **Edit** > **Roles**.
15. Select the check box next to the name of the role, then click **Enable**.
16. Click **OK**.
17. Update the Identity Server.

Enabling and Disabling Role Policies

In order for a role policy to function, you must enable it for the Identity Server configuration.

1. In the Administration Console, click **Devices > Identity Servers** > **Edit > Roles**.
2. Select the role policy's check box, then click **Enable**.
3. To disable the role policy, select the role policy's check box, then click **Disable**.
4. After enabling or disabling role policies, update the Identity Server configuration on the **Servers** tab.

Importing and Exporting Role Policies

You can import and export role policies in order to run them in other Identity Server configurations. When you import a role, ensure that you have enabled any Liberty profile that is referenced in the role policy, in order to correctly display the policy in the interface. However, the policy still evaluates if you have not enabled the profile.

You must also enable roles after importing them to an Identity Server configuration. See "Enabling and Disabling Role Policies" on page 642. Click **Devices > Identity Servers > Edit > Roles**.

When you export a role policy, the system saves it as a `.txt` file at the location of your choosing. After you import a role policy, you must update the Identity Server configuration.

To export a role policy:

1. In the Administration Console, click **Policies > Policies**.
2. Select a policy, then click **Export**.
3. (Optional) Modify the name suggested for the file.
4. Click **OK**.
5. Using the features of your browser, specify where you want the file to be copied.
6. Click **OK**.

To import a role policy:

1. In the Administration Console, click **Policies > Policies**.
2. Click **Import**, then browse to and select the file.
3. Click **OK**.
4. When the policy appears in the list, click **Apply Changes**.

Authorization Policies

Authorization policies are used when you want to protect a resource based on criteria other than authentication, and you want Access Manager Appliance to enforce the access restrictions. Authorization policies are enforced when a user requests data from a resource.

Access Manager Appliance supports Access Gateway Authorization policies for protecting resources of the Access Gateway.

The first step in creating an Authorization policy is determining the criteria for restricting access. The second step is translating those criteria into rules and conditions for a policy. This section describes the policy elements, but your resource and your security requirements determine which elements to use when creating the policy.

- “Designing an Authorization Policy” on page 643
- “Creating Access Gateway Authorization Policies” on page 653
- “Sample Access Gateway Authorization Policies” on page 655
- “Conditions” on page 662
- “Importing and Exporting Authorization Policies” on page 691

Designing an Authorization Policy

When you create an Authorization policy, you need to configure one or more rules. Each rule consists of two parts: (1) one or more conditions the user must meet and (2) the action to perform when the user meets the conditions or does not’ meet the conditions. The action can be to either allow or deny access to the resource. This section describes how to use the following elements when creating a policy:

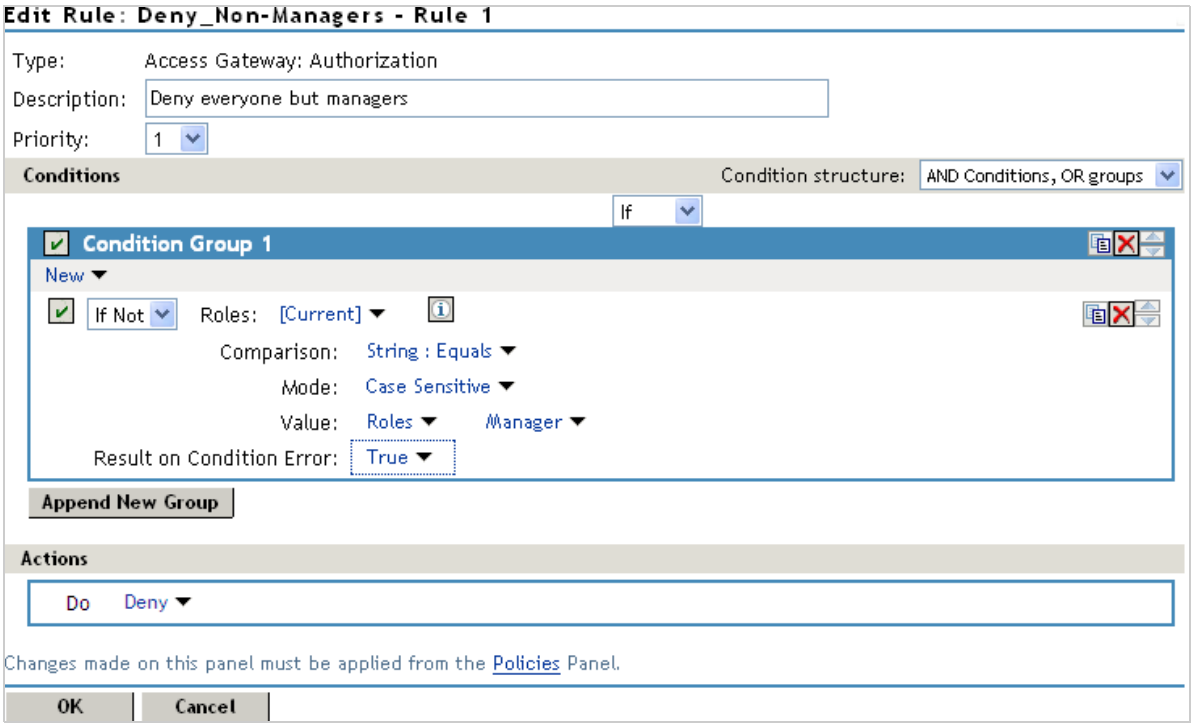
- “Controlling Access with a Deny Rule and a Negative Condition” on page 643
- “Configuring the Result on Condition Error Option” on page 644
- “Many Rules or Many Conditions” on page 645
- “Using Multiple Conditions” on page 645
- “Controlling Access with Multiple Conditions” on page 647
- “Using Permit Rules with a Deny Rule” on page 648
- “Using Deny Rules with a General Permit Rule” on page 649
- “Public Policies” on page 651
- “General Design Principles” on page 651
- “Using the Refresh Data Option” on page 651
- “Assigning Policies to Resources” on page 652

Controlling Access with a Deny Rule and a Negative Condition

To deny access to the correct set of users, you need to know the characteristics of the users you don’t want to access the resource, as well as the characteristics of the users you do want to access the resource.

Some very simple policies can be created by using a Deny action. For example, suppose you have an application that you only want managers to access. If you have set up a role that assigns all managers to the Manager role, you can use this characteristic for an Authorization policy. Such a rule would be similar to the following:

Figure 6-15 *Simple Rule*



This rule evaluates the user, and if the user does not belong to the Manager role, the user matches the condition. The action for matching the condition is to deny access. The managers, who belong to the Manager role, do not match the condition and the Deny action is not applied to them.

The **Result on Condition Error** option is set to True. You don't want an error to cause the policy to assume that the user is a manager. If an error occurs, you want the policy to assume that the user is not a manager, so he or she matches the condition and the Deny action is applied.

Configuring the Result on Condition Error Option

The **Result on Condition Error** option allows you to specify what the condition returns when the comparison of the two values returns an error rather than the results of the comparison. Select either **False** or **True**. You need to analyze the logic of your policy carefully, because if you set up this option incorrectly, error conditions can allow access to a resource. Consider the following:

- If your rule is a Permit rule and you do not want the action applied when an error occurs, select **False** for this option.
- If your rule is a Deny rule with an **If Not** condition and you want the action applied when an error occurs, select **True**.

Many Rules or Many Conditions

You can design your policy to have many rules with a single condition and action, or you can design your policy to have fewer rules, with each rule containing many conditions.

For example, suppose you have a resource that you don't want users accessing on Monday, Wednesday, and Friday between 1:00 a.m. and 2:00 a.m. You could set up three rules, one for each day, or you could set up one rule with three conditions. If all the conditions have the same action (for example, deny access with the same reason), it is simpler to put them in the same rule. However, if you have a customized message to return for each day, you need to put them in separate rules.

Each rule contains the following:

- Zero or more conditions. A condition specifies how the request data is evaluated for a True or False match. Conditions are evaluated in the order in which they are listed.
- One or more condition groups. Conditions are placed in condition groups, which gives you the flexibility of creating a policy that allows the user to match the conditions in one group but not the conditions in the other condition groups. Or you can set up the condition groups to require that the user matches at least one condition in each condition group.
- An action, which grants access, denies access, or redirects the users.

Conditions, conditions groups, and the interaction among them allow you to create very simple rules (if A, then grant access) to very complex rules (if A, B, and C, but not D and E, then grant access).

Using Multiple Conditions

The **Condition structure** option controls how conditions within a condition group interact with each other and how condition groups interact with each other. Select one of the following:

- **AND Conditions, OR groups:** If the conditions are ANDed, the user must meet all the conditions in a condition group to match the profile. If the condition groups are ORed, the user must meet all of the conditions of one group to match the profile. This option allows you to set up two or more profiles into which a user could fit and be considered a match. For example, suppose you create the following Permit rule:

  The first condition group contains the following conditions:

  1. The user's department must be Engineering.
  2. The request must come on a weekday.

  The second condition group contains the following conditions:

  1. The user's department must be Information Services and Technology (IS&T).
  2. The request must come on a weekend.

  With this rule, the engineers who match the first condition group have access to the resource during the week, and the IS&T users who match the second condition group have access to the resource on the weekend.
- **OR Conditions, AND groups:** If the conditions are ORed, the user must meet at least one condition in the condition group to match the profile. If the conditions groups are ANDed, the user must meet at least one condition in each condition group to match the profile. For example, suppose you create the following allow rule:

  The first condition group contains the following conditions:

  1. The user's department is Engineering.
  2. The user's department is Sales.

The second condition group contains the following conditions:

1. The user has been assigned the Party Planning role.
2. The user has been assigned the Vice President role.

With this rule, the Vice Presidents of both the Engineering and Sales departments can access the resource, and the users from the Engineering and Sales department who have been assigned to the Party Planning role can access the resource.

At the top of each condition group, there is an option that allows you to control whether the user must match the conditions to match the profile or whether the user matches the profile if the user does not' match any of the conditions. Depending upon your selection for the Condition structure, you can select from the following:

- If/If Not
- Or/Or Not
- And/And Not

Conditions also have similar Not options, so that a user can match a condition by not matching the specified value.

Adding Multiple Conditions

To add another condition to a condition group, click **New**, then select a condition. To copy an existing condition, click the **Copy Condition** icon  . New conditions are always added to the end of the condition group. Use the **Move**  buttons to order the conditions in the condition group.

Adding New Condition Groups

To add another condition group to the rule, click **Append New Group**. To copy the existing condition group, click the **Copy Group** icon  . New condition groups are always added to the end to the Conditions section. Use the **Move**  buttons to order the condition groups.

Disabling or Moving Conditions and Condition Groups

Condition groups and conditions within them can be disabled by clicking the Enabled check mark  , which changes the icon to the **Disabled** icon .

You usually disable a condition or condition group when testing a new rule, and if you decide that the condition or condition group is not needed, you can then use the **Delete**  button to delete the condition or condition group from the rule. Use the **Move**  buttons next to the **Delete** button to move a condition up or down within its group. Condition groups also have **Move** buttons.

Controlling Access with Multiple Conditions

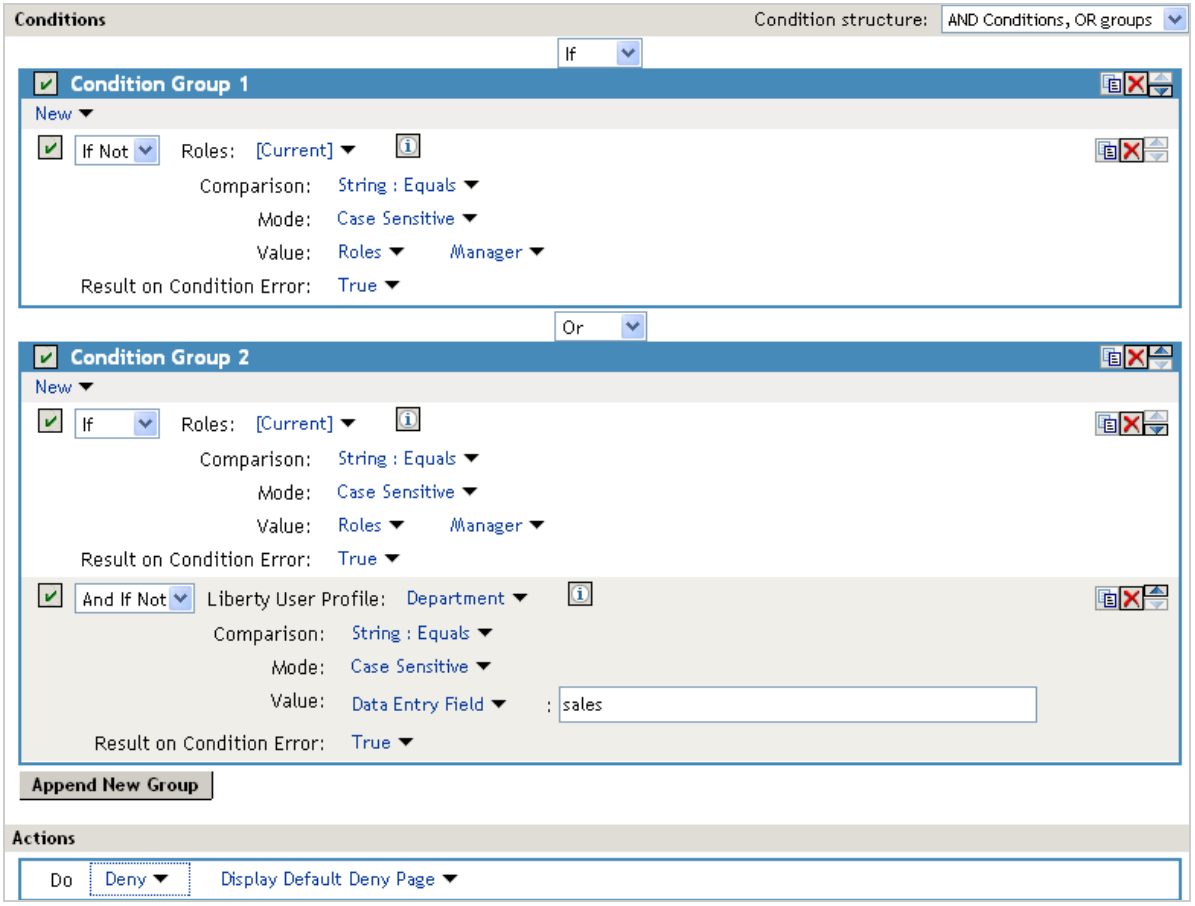
A policy requires multiple conditions when you have more than one required condition for granting access. For example, suppose you can easily identity your managers because they have all been assigned the role of Manager, and you have a resource that only the sales managers must access. Such a policy requires two conditions for granting access: the Manager role and membership in the sales department. For a Deny rule, the rule needs two condition groups:

- The first condition group matches all users who are not managers. This causes the Deny action to be applied.
- The second condition group matches the users who are managers but don't belong to the sales department. Because they match both conditions, the Deny action is applied. For these two condition groups to work with this logic, the **Condition structure** is set to **AND Conditions, OR groups**.

The users who are managers and who belong to the sales department do not match either condition group. The Deny action is not applied, and they are allowed access.

Such a rule would look similar to the following:

Figure 6-16 *A Rule with Two Condition Groups*



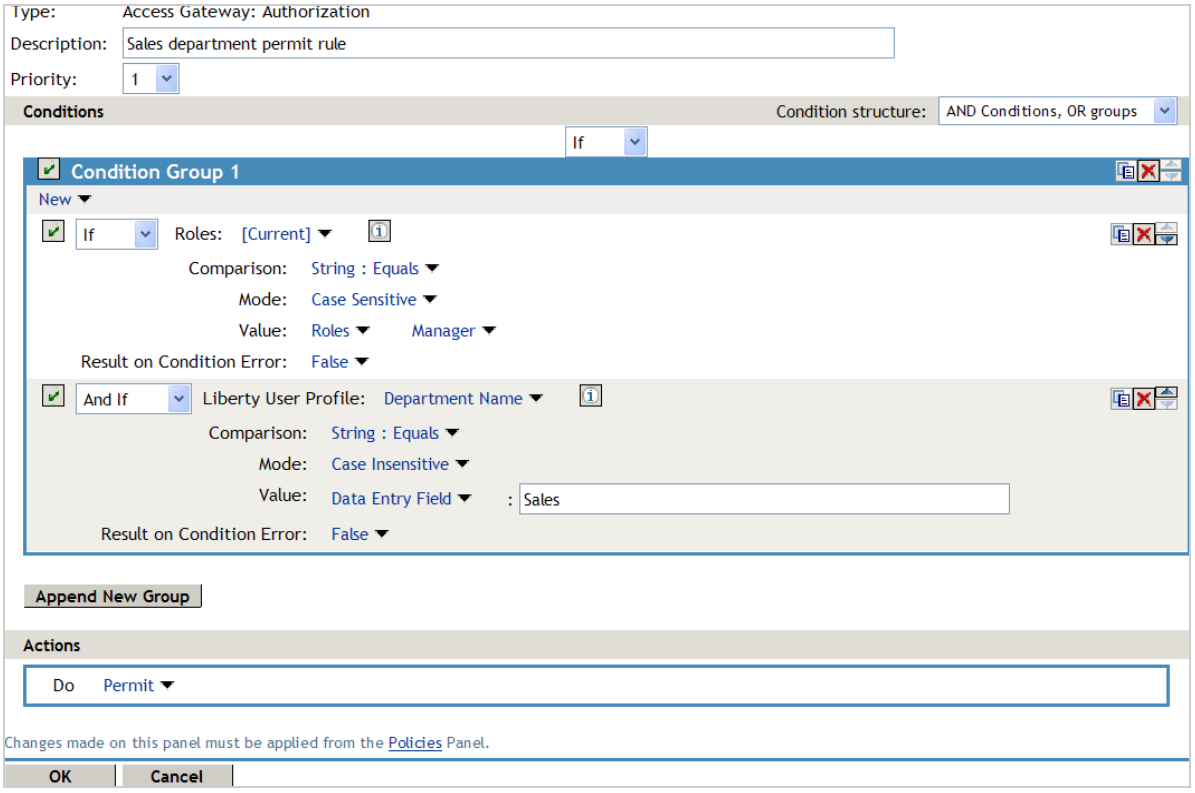
This second condition group could be implemented as the second rule of the policy. If so, it must be set as a lower priority than the first rule. Because most systems would have more users than managers, the user rule would be used more frequently, so it must come first.

Using Permit Rules with a Deny Rule

You can also create policies that contain one or more Permit rules and then create the lowest priority rule in the policy as a Deny rule with no conditions. In such a policy, as soon as an allow match is processed, the rest of the rules are not processed and the user is granted access to the resource. The Deny rule is only processed if the user does not match one of the allow rules, and because all users match a rule with no conditions, the user is denied access to the resource.

The first rule in such a policy for the sales application would look similar to the following.

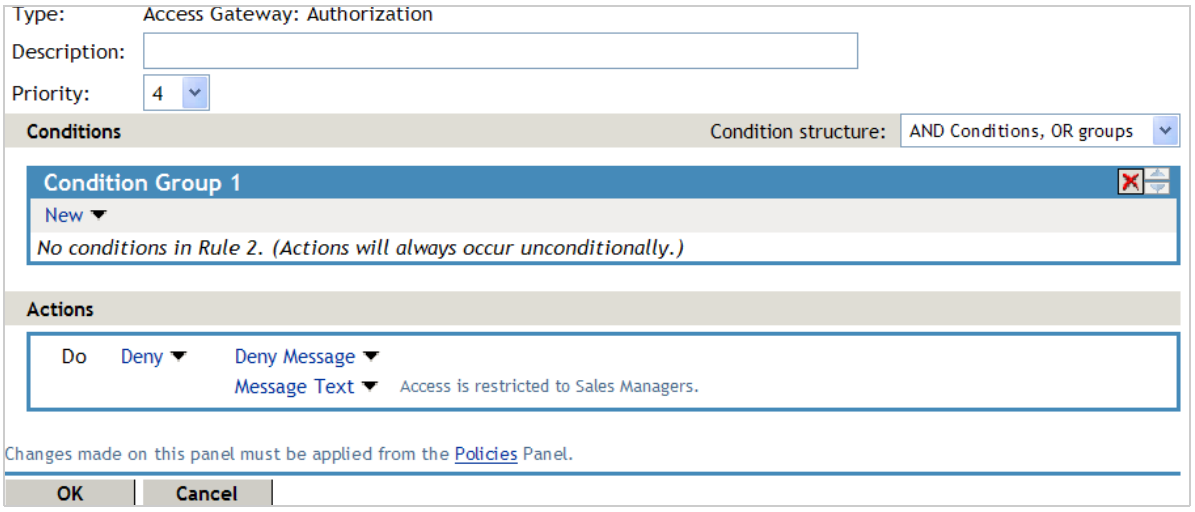
Figure 6-17 *Rule 1 Granting Access*



The conditions in Rule 1 are ANDed, which requires the user to match both conditions before they are granted access to the resource. The priority is set to 1, so this rule is the first rule that the Access Gateway processes.

The second rule would look similar to the following:

Figure 6-18 *Rule 2 Denying Access*



Because this rule has no conditions, any user who does not match the first rule does match this rule and is denied access. The priority of this rule is set lower than the Permit rule so that the Permit rule is processed first.

Using Deny Rules with a General Permit Rule

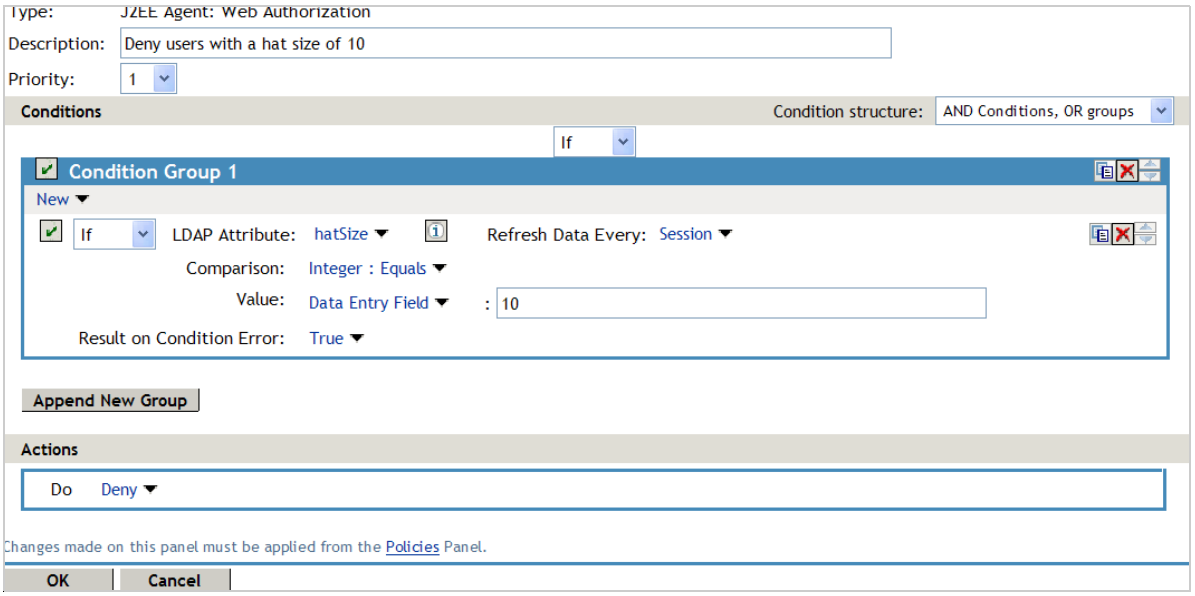
You can also create policies that contain one or more Deny rules and then create the lowest priority rule in the policy as a Permit rule with no conditions. In such a policy, as soon as a Deny rule matches a user, the rest of the rules are not processed and the user is denied access to the resource. The Permit rule is only processed if the user does not match one of the Deny rules. Because all users match a rule with no conditions, the user is allowed access to the resource.

The key to creating this type of policy is making sure all the Deny rules match the users you do not want accessing the resource and making sure that the **Result on Error Condition** option is set correctly.

For example, suppose one of the Deny rules uses an LDAP attribute for the condition and that the attribute is a hatSize attribute. Some of your users do not have a hatSize attribute, so when they access the resource, the comparison generates an error. If **Result on Error Condition** option is set to False, the action (Deny) is not applied, and the next rule in the policy is processed. If that rule is the general Permit rule, then they are allowed access to the resource because they experienced an error. To prevent this behavior, you need to set the **Result on Error Condition** option to True, so that the Deny action is applied. Your rule then denies access to everyone whose hatSize attribute matches the specified value and everyone who does not have the attribute.

The Deny rule for such a policy would look similar to the following:

Figure 6-19 *Deny Rule Configured for Error Conditions*



For most people, Deny rules are harder to write than Permit rules. You not only need to carefully configure the **Result on Condition Error** option, you must also carefully consider the consequences of the condition not matching a user. When a user does not' match the condition, the Action is not applied and the next rule in the policy is evaluated. For example, suppose the URL condition is set to the compare the following value:

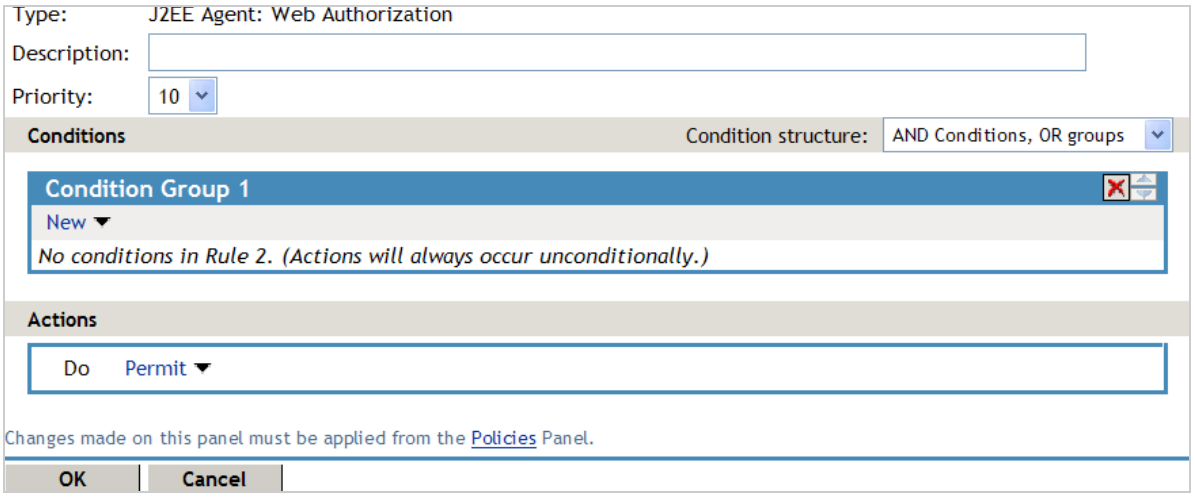
`http://sales.provo.novell.com/meetings/?`

If the URL in the request is `http://sales.provo.novell.com/meetings/january`, the user does not match the condition, because the ? applies only to the files in the `meetings` directory and not to the subdirectories. The Action is not applied, and the next rule or policy is evaluated. Consider the following possibilities:

- If you want the condition to match all files and subdirectories, you need to change the ? wildcard to the wildcard.
- If you want the condition to allow access to the files in the `/meetings` directory but deny access to the subdirectories, you need to negate the condition so it evaluates as follows: if the URL is not a request for the `/meetings/?` directory, deny access. If you select this type of condition, you need to set the **Result on Condition Error** option to True. If the comparison returns an error and there is the possibility that the request is for a subdirectory, you want the user to be denied access.

The general Permit rule for a Deny policy would look similar to the following:

Figure 6-20 *General Permit Rule*



Public Policies

You can create public authorization policies, which are policies that apply to everyone, by leaving the **Condition** section empty. In the **Action** section, you specify either to deny or to permit access to the resource. Then you assign the policy to the protected resource.

General Design Principles

When you design a policy, remember the following principles:

- Logged-in users are allowed access to a protected resource unless the policy denies access.
- Priority determines the order in which rules are applied.
- The Conditions section of the rule must evaluate to True in order for the Action section to be applied. If the Condition section evaluates to False, the Action section is ignored and the policy moves to the next rule. If another rule does not exist, the user is granted access to the resource.
- Rules are only processed until a user matches the conditions in a rule and its action is applied. If a user matches the first rule in a policy, that action is applied, and the rest of the rules in the policy are ignored.
- If two rules have the same priority, Deny rules are applied before Permit rules.
- After you have designed your policy, created it, and assigned it to a resource, you need to test the policy. You need to log in as the type of user who must be granted access, as the type of user who must not be granted access, and as a user who generates an error on condition evaluation.

Using the Refresh Data Option

Authorization policies are processed when a user requests access to a resource. The results and the values of the data items are cached for the user session. This means that when the user requests a second time to access the resource, the policy is evaluated, but the data values from the first

evaluation are used. When a data item is cached for the user session, the user must log out and log back in to trigger new data values. (For information about how long the data items are cached, see “The Policy Is Using Old User Data” on page 1007.)

The LDAP Attribute can be configured to refresh its value according to a specified interval. This means the attribute value is read not just on the first request that triggers the policy evaluation, but when the interval expires. You can select to cache the value for the user session, the current request, or a time interval varying from 5 seconds to 60 minutes.

If the requested page contains links, you must usually cache the data for more than a single request. Each link on the page generates a new request.

You can use this feature for situations when you do not want to force the user to log in again to gain rights to resources or to revoke rights to resources. For example, suppose that you have an Authorization policy that grants access based on an LDAP attribute having a “yes” value. Users with a “no” value in this attribute are denied access.

If you don’t enable the Refresh Data option on this attribute in the policy condition, the policy is evaluated when the user first tries to access the resource. The value for the attribute is cached for the user session, and until the user logs out, that is the value that is used.

However, if you enable the Refresh Data option on this attribute in the policy condition, the policy is evaluated when the user first tries to access the resource. When the user sends a second request to access the resource and the cached value has been marked old, the Refresh Data option causes the value of the attribute to be read again from the LDAP server. This new value is used to evaluate the policy and any other policy that is triggered by the request.

- If the value from the first request to the second request changes from no to yes, the user gets access to the resource.
- If the value from the first request to the second request changes from yes to no, the user is denied access to the resource.

For example:

- If the attribute controls access to employee resources and an employee leaves, a quick change of this attribute value cuts the employee off from the resources that must be available to employees only.
- If the attribute controls access to a software download site and a user has just purchased a product, a quick change to this attribute value can grant access to the download site.

IMPORTANT: This feature needs to be used with caution. Because querying the LDAP server slows down the processing of a policy, LDAP attribute values are normally cached for the user session. Enable this option only on those attributes that are critical to the security of your system or to the design of your work flow.

Assigning Policies to Resources

For information about how to assign the Access Gateway policy, see “Assigning an Authorization Policy to a Protected Resource” on page 107.

Creating Access Gateway Authorization Policies

An Authorization policy specifies conditions that a user must meet in order to access a resource or to be denied access to a resource. The Access Gateway enforces these conditions.

To create an Authorization policy:

1. In the Administration Console, click **Policies > Policies**.
2. Select the policy container, then click **New**.
3. Specify a name for the policy, then select **Access Gateway: Authorization** for the type of policy.
4. Fill in the following fields:

   Description: (Optional) Describe the purpose of this rule.

   Priority: Specify the order in which a rule is applied in the policy, when the policy has multiple rules. The highest priority is 1 and the lowest priority is 10. If two rules have the same priority, a Deny rule is applied before a Permit rule.
5. In the **Condition Group 1** section, click **New**, then select one of the following:
   - **Authentication Contract:** Allows you to control access based on the contract the user used for login. For configuration information, see “Authentication Contract Condition” on page 663.
   - **Client IP:** Allows you to control access based on the IP address of the client making the request. For configuration information, see “Client IP Condition” on page 665.
   - **Credential Profile:** Allows you to control access based on the credentials the user specified during authentication. For configuration information, see “Credential Profile Condition” on page 666.
   - **Current Date:** Allows you to control access based on the date of the request. For more information, see “Current Date Condition” on page 667.
   - **Day of Week:** Allows you to control access based on the day the request is made. For configuration information, see Day of Week Condition.
   - **Current Day of Month:** Allows you to control access based on the month the request is made. For configuration information, see “Current Day of Month Condition” on page 670.
   - **Current Time of Day:** Allows you to control access based on the time the request was made. For configuration information, see “Current Time of Day Condition” on page 670.
   - **HTTP Request Method:** Allows you to control access based on the request method. For configuration information, see “HTTP Request Method Condition” on page 672.
   - **LDAP Attribute:** Allows you to control access based on the value of an LDAP attribute. For configuration information, see “LDAP Attribute Condition” on page 673.
   - **LDAP OU:** Allows you to control access based on the value of an LDAP organizational unit. For configuration information, see “LDAP OU Condition” on page 676.
   - **Liberty User Profile:** Allows you to control access based on the value of a Liberty attribute. For configuration information, see “Liberty User Profile Condition” on page 677.
   - **Roles:** Allows you to control access based on the roles a user has been assigned. For configuration information, see “Roles Condition” on page 678.
   - **Risk Score:** Allows you to define a condition group as part of the authorization policy that uses the risk score from Identity Server to protect a resource. For more information, see “Risk Score” on page 679
   - **URL:** Allows you to control access based on the URL in the request. For configuration information, see “URL Condition” on page 680.

- **URL Scheme:** Allows you to control access based on the scheme in the URL of the request (for example, HTTP or HTTPS). For configuration information, see "URL Scheme Condition" on page 681.
- **URL Host:** Allows you to control access based on the hostname in the URL of the request. For configuration information, see "URL Host Condition" on page 682.
- **URL Path:** Allows you to control access based on the path in the URL of the request. For configuration information, see "URL Path Condition" on page 683.
- **URL File Name:** Allows you to control access based on the filename in the URL of the request. For configuration information, see "URL File Name Condition" on page 685.
- **URL File Extension:** Allows you to control access based on the file extension in the URL of the request. For configuration information, see "URL File Extension Condition" on page 686.
- **Virtual Attribute:** Allows you to control access based on the value of the virtual attribute. For configuration information, see "Virtual Attribute Condition" on page 687.
- **X-Forwarded-For IP:** Allows you to control access based on the value in the X-Forwarded-For IP header of the HTTP request. For configuration information, see "X-Forwarded-For IP Condition" on page 688.
- **Condition Extension:** (Conditional) If you have loaded and configured an authorization condition extension, this option specifies a condition that is evaluated by an outside source. This outside source returns either True or False. See the documentation that came with the extension for information about what is evaluated.
- **Data Extension:** (Conditional) If you have loaded and configured an authorization data extension, this option specifies the value that the extension retrieves. You can then select to compare this value with an LDAP attribute, a Liberty User Profile attribute, a Data Entry Field, or another Data Extension. For more information, see the documentation that came with the extension.

6 To add multiple conditions to the same rule, either add a condition to the same condition group or create a new condition group. For information about how conditions and condition groups interact with each other, see "Using Multiple Conditions" on page 645.

7 In the **Actions** section, select one of the following:

- **Permit:** Allows the user to access the resource.
- **Redirect:** Specify the URL to which you want users redirected when they meet the conditions of this policy.
- **Re-authenticate with Contract:** Select the action to be performed after execution of the rule. If you select **Re-authenticate with Contract**, select the contract to be used.

NOTE: If **Redirect** is configured as an Authorization policy action and you attempt to configure **Re-authenticate with Contract** option instead of **Redirect**, no contracts are displayed.

To workaround this issue, Select **Permit** or **Deny** and then select **Re-authenticate with Contract**. The list of contracts are displayed

---

- **Deny:** Select one of the following deny actions:

  Display Default Deny Page: Displays a generic message, indicating that the user has insufficient rights to access the resource.

  Deny Message: Allows you to provide a customized message that is displayed to users who are denied access.

  Redirect to URL: Allows you to specify a URL that users are redirected to when they are denied access. For example:

```
http://www.novell.com
```

- **Action Extension (Permit):** Select an action from the list of permit extensions. This action permits access to the resource and performs the additional action that the extension is designed to perform. If an action extension is not available, see “Adding Policy Extensions” on page 598 for information about uploading, configuring, and importing extensions.
- **Action Extension (Deny):** Select an action from the list of deny extensions. This action denies access to the resource and performs the additional action that the extension is designed to perform. If a deny extension is not available, see “Adding Policy Extensions” on page 598 for information about uploading, configuring, and importing extensions.

8 (Conditional) If you have installed an action obligation extension, you can click **New** in the **Actions** section, and select the action. This causes the extension to perform whatever action it is designed to perform whenever a user matches the conditions of this rule. This type of action is usually always configured in addition to a permit or deny action. If the obligation option is not available, see “Adding Policy Extensions” on page 598 for information about uploading, configuring, and importing extensions.

9 To save the rule, click **OK**.

10 To add another rule, click **New** or to save the policy, click **OK**, then click **Apply Changes**.

11 Assign the policy to a protected resource (see “Assigning an Authorization Policy to a Protected Resource” on page 107).

Sample Access Gateway Authorization Policies

- “Sample Policy Based on Organizational Rules” on page 655
- “Sample Workflow Policy” on page 658

Sample Policy Based on Organizational Rules

The following sections describe a scenario with an organizational division, then describe two types of policies that enforce the requirements of the scenario:

- “Company Scenario” on page 655
- “LDAP Context Policies” on page 656
- “Role Policies with Authorization Policies” on page 657

Company Scenario

Suppose that the company LDAP directory has the following organization:

ou=sales,o=acme
ou=dev,o=acme
ou=hr,o=acme

Suppose that this company has the following configuration and requirements:

- Under each branch of the tree, the system administrator has created the users who work in these departments.
- Each department has its own Web resources, and other departments must be denied access to these resources.

With this type of configuration, you can use the LDAP context condition to create authorization policies or you can create role policies that are used in conjunction with authorization policies.

LDAP Context Policies

With such an organization, you can create a policy that either allows or denies access based on the LDAP context of the user's DN. You can use the LDAP context of the user DN to separate the users into their departments and then grant access based on the context match. You need to create protected resources for the Web resources of the department, create a policy for each protected resource, and assign a policy to the protected resources.

The following procedure explains how to configure such a policy for the sales department.

1. Click **Policies** > **Policies** > **New**, specify a name for the policy, select **Access Gateway: Authorization** as the type, then click **OK**.
2. For **Condition Group 1**, click **New**, then select **Credential Profile**.
3. Fill in the following fields:

   LDAP Credentials: Select **LDAP User DN**.

   If/If Not: Select **If Not**.

   Comparison: Select **Contains Substring**.

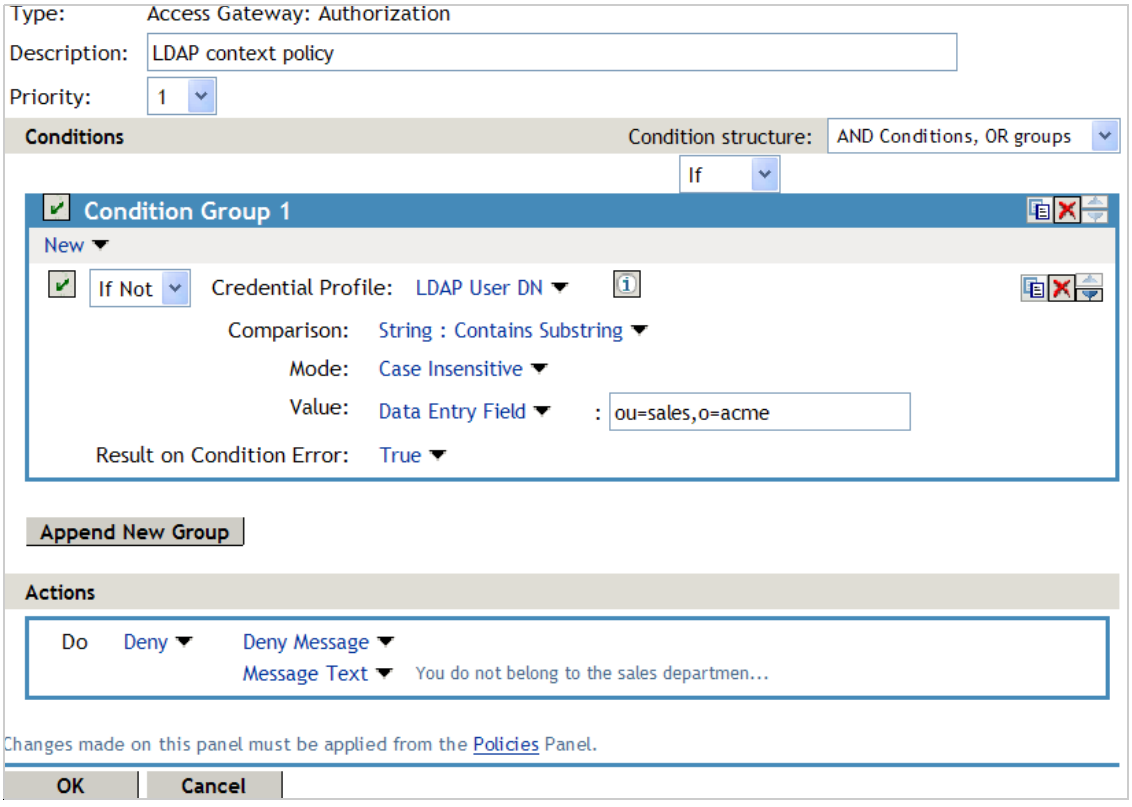
   Mode: Select **Case Insensitive.**

   Value: Select **Data Entry Field**. In the text box, type the following value:

   ```
   ou=sales,o=acme
   ```

   Result on Condition Error: Select **True**.
4. In the **Actions** section, select **Deny**.

   Your policy must look similar to the following:



This sets up the condition so that the following occurs:

- When the user does not belong to the sales department, the user is denied access.
- When the user belongs to the sales department, the user is granted access.
- When an error occurs evaluating the conditions in the rule, the user is denied access.

5 Assign the policy to the protected Web resources of the sales department (see “Assigning an Authorization Policy to a Protected Resource” on page 107).

6 Repeat these steps for the other two departments, changing the **Value** field to match the appropriate department.

Role Policies with Authorization Policies

Because of the company’s organization, you need to create three role policies, one for the sales users, one for the development users, and one for the human resource users. You can then use these roles as conditions in authorization policies to allow and deny access. The first time you use roles in an authorization policy, there is extra setup because you must create the role policies. However, after the role policies are created, you can use them in multiple authorization policies.

The following instructions explain how to use the Sales role to create a policy that controls access to a protected resource. For instructions on how to create the Sales role, see “Creating a Role by Using the Location of the User Objects” on page 634.

You need to decide on the type of Authorization policy you want to create. For example, you can create a Deny policy that denies access to everyone who does not match the condition (in this case, the Sales role).

Or you can create a two-rule policy that allows access to everyone that matches the condition. The first rule grants access to everyone who has the Sales role, and the second rule denies access to everyone who did not match the conditions of the first rule. (Other methods are also possible.) Because the proposed Deny policy is very similar to the LDAP Context Policies example, the following procedures explain how to create the two-rule policy.

1 In the Administration Console, click **Policies > Policies** > **New**.

2 Specify a name for the policy, select **Access Gateway: Authorization** as the type, then click **OK**.

3 (Optional) Provide a description for the rule.

4 In **Condition Group 1**, click **New**, and select **Roles**.

5 Fill in the following fields:

If/If Not: Select **If**.

Roles: Select **[Current]**.

Comparison: Select **String: Equals**.

Mode: Select **Case Insensitive**

Value: Select **Roles,** then select **Sales.**

Result on Condition Error: Select **False**.

6 Under **Actions**, select **Permit**, then click OK.

These steps create the Permit rule and set up the condition so that the following occurs:

- When the user does not match the condition because the user does not belong to the Sales role, the policy engine moves to the next rule in the policy.

- When the user does match the condition because the user belongs to the Sales role, the user is granted access.
- If an error occurs when evaluating the condition of the policy, the user does not match the condition and the policy engine moves to the next rule in the policy.

7 In the **Rule List**, click **New**.

This second rule is for denying access to everyone who does not match the condition in Rule 1. Processing of the policy stops when a user matches a rule; therefore all users who match Rule 1 are granted access and the policy engine does not evaluate the second rule.

8 Set the **Priority** to be 2 or greater.

You want the Permit rule to be processed first, so it must have a priority of 1. The Deny rule needs to be processed last, so it needs a lower priority than the Permit rule.

9 Leave the **Condition Group 1** empty.

The **Conditions** section is left empty so that everyone who does not match the conditions of the Permit rule is denied access to the resource.

10 In the **Actions** section, select **Deny** and either accept the default action or select one of the other actions.

11 Click **OK** twice.

12 Click **Apply Changes** on the Policies page.

13 Assign the policy to the protected Web resources of the sales department (see “Assigning an Authorization Policy to a Protected Resource” on page 107).

Sample Workflow Policy

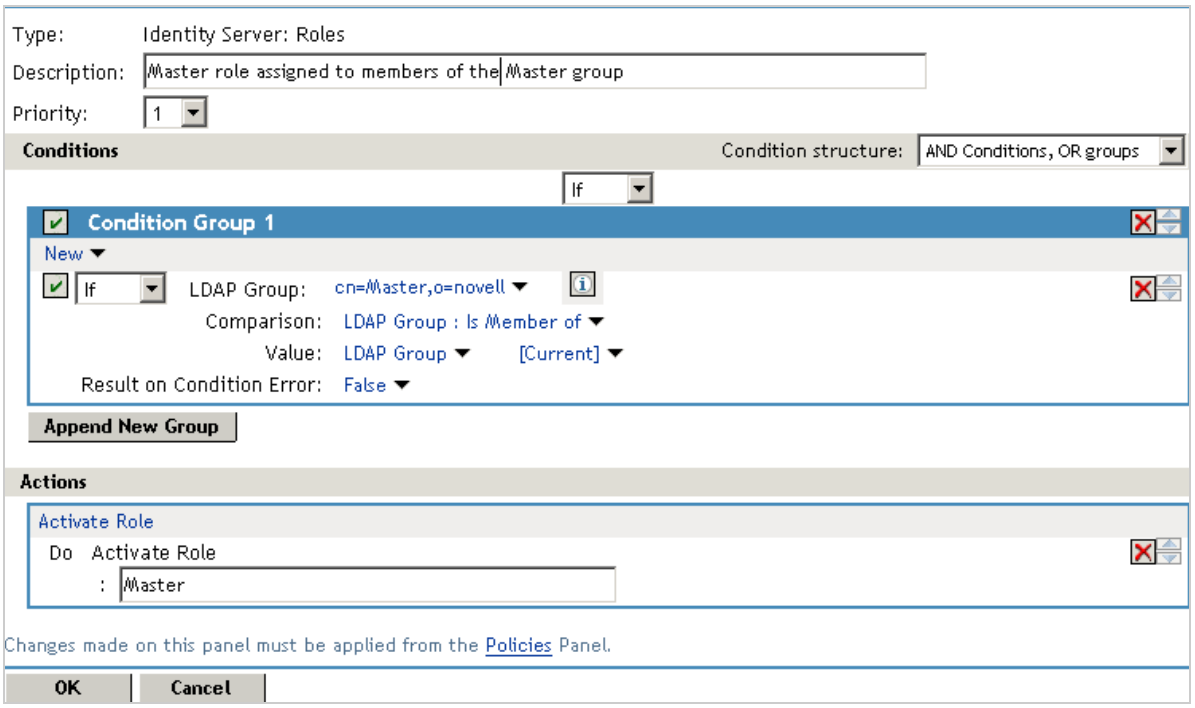
One of the common workflow problems that an Authorization policy can solve is what to do with users who are denied access to resource. Most of the time they have a legitimate reason for trying to access the resource and need contact information to request access to the resource. You can add this contact information to a Web page and redirect the users to this page when the policy denies the user access.

To create such a workflow, you need to create an HTML page with the necessary information for making the request for access. It can be as simple as a contact name or it can be an actual form that the user submits to the organization that controls access to the resource.

You then need to create an Authorization policy that redirects the denied users to this page. The following sample policy uses a role for the access condition, but the same workflow can be created using any of the other conditions available for an Authorization policy. For this example, assume that the user is granted a Master role if the user is a member of the Master group. The organization that controls access to the resource is the owner of the Master group and can add and delete members from the group. When the owner of the Master group receives a request for access to the resource, the owner can evaluate the user, and if the user meets their standards, the owner adds the user to the Master group.

You can use the Master group to create an Access Manager Appliance Role policy. This policy for the Master role must look similar to the following:

Figure 6-21 *A Role Policy with an LDAP Group Condition*

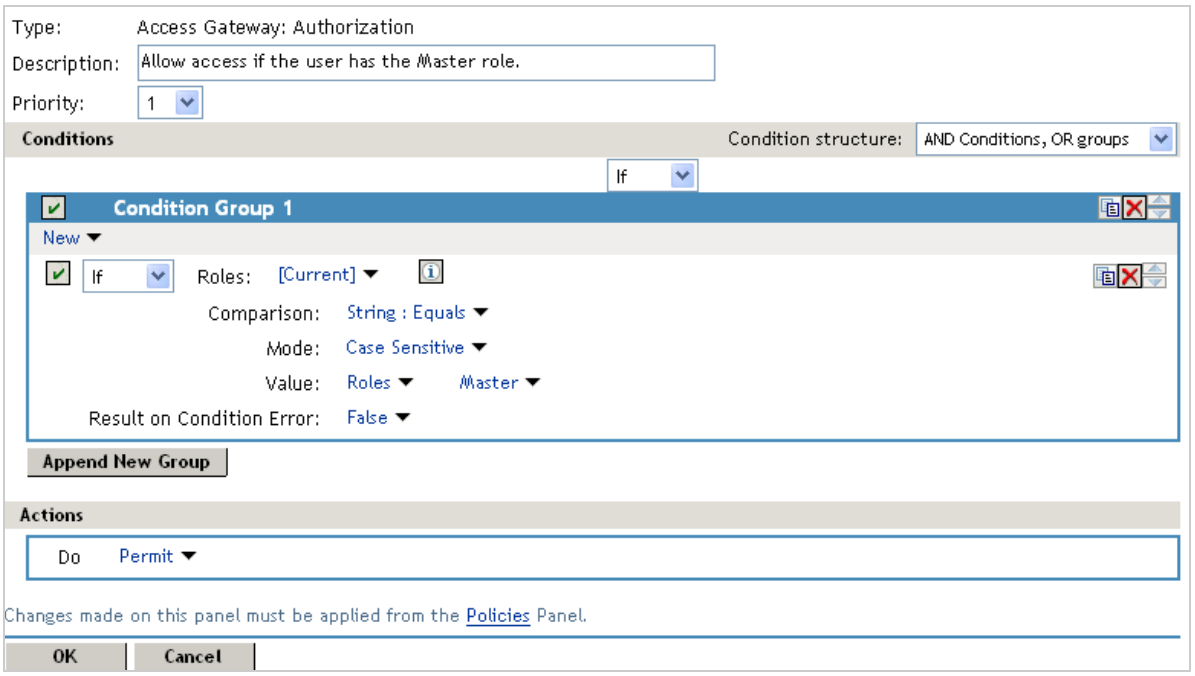


This rule grants the user the Master role if the user belongs to the cn=Master,o=novell LDAP group. If the user does not' belong to this group or if an error occurs trying to get the data, the user is not assigned the role. This occurs because both the condition and the **Result on Condition Error** evaluate to False, which prevents the Action from being applied.

After creating the Role policy, apply the changes and enable the Role for the Identity Server.

You can then use this role to create an Authorization policy that contains two rules. The first rule grants access to the users who have the Master role (and are therefore members of the Master group). This rule must look similar to the following:

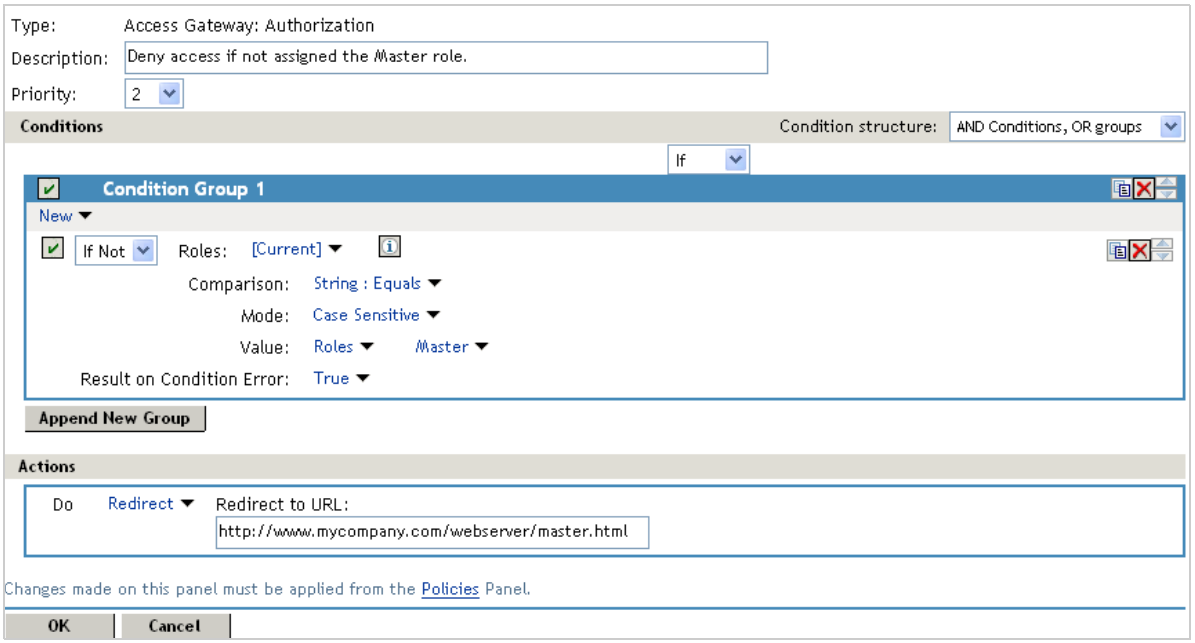
Figure 6-22 *A Permit Rule with a Role Condition*



This rule permits users who are assigned the Master role to have access to the resource. If the user does not match the condition or if an error occurs accessing the user's role information, the user is sent to the next rule because both the condition and the **Result on Condition Error** evaluate to False.

The second rule in the policy must deny access to those who are not assigned the Master role and must redirect them to the page where they can request access. You can do this with a rule that checks to see if they are assigned the Master role. In this type of rule, the condition needs to be an **If Not** condition.

Figure 6-23 *A Deny Rule with a Redirect URL*



With an **If Not** condition, the condition evaluates to True when the user does not match the condition. With such a rule, you want the **Result on Condition Error** to also evaluate to True. If there is an error obtaining role information for the user, you don't want the rule to assume that the user had the Master role. You want the rule to assume that the user had no roles, or in other words, you want the error condition to evaluate to True.

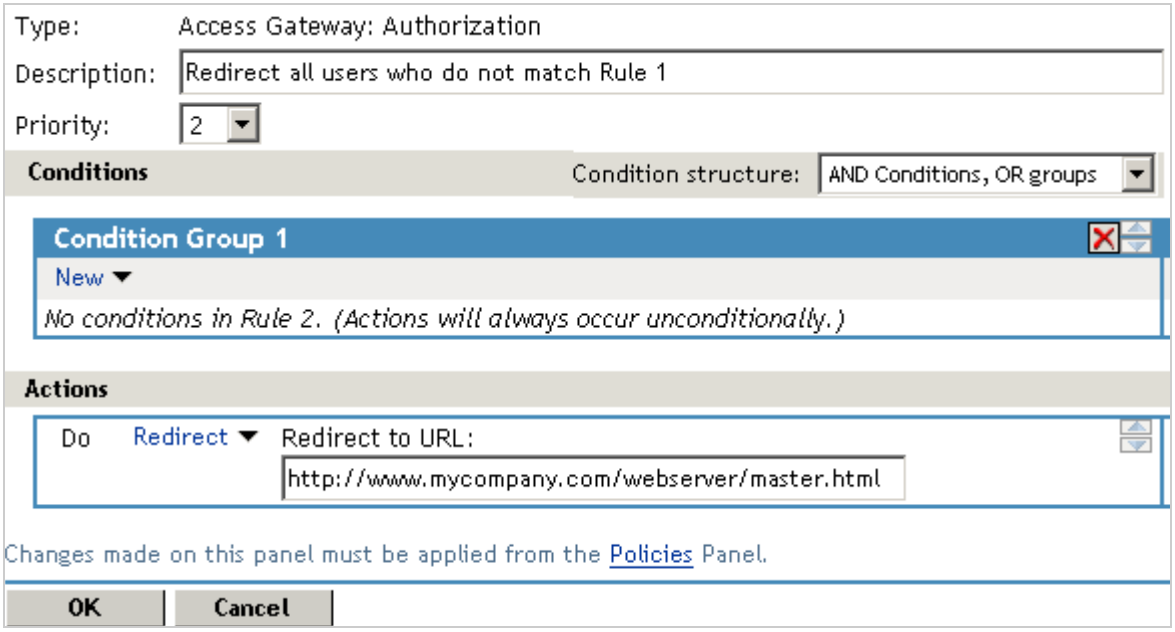
Because the condition evaluated to True, the Action is applied to the user. The value specified in the **Redirect to URL** text box must specify the page that contains the information about how to request access.

This redirect rule could be the only rule in the Authorization policy, because the users who are assigned to the Master role do not match the rule and are thus allowed access. Having the first rule that grants access because they have the Master role just makes the logic of the policy clearer.

If you create the first rule that grants users with the Master role access, you can use a general Deny rule for the second rule.

It must look similar to the following:

Figure 6-24 *A General Deny Rule*



A general Deny rule has no conditions, so it matches everyone that does not match the first rule in the policy. You can add more rules to this policy to tighten security so that not all users are redirected to the site that contains the information about how to request access. For this type of policy, the last rule would be a general Deny rule with no conditions and without a redirect. The rules between Rule 1, which granted access to people assigned to the Master role, and the last rule, which denies everyone, must be rules that identify the types of users who have legitimate reasons for requesting access, and these rules must contain the redirect action.

After you have saved the Authorization policy, you need to assign it to the protected resource or resources that require the Master role, then update the Access Gateway.

Conditions

This section describes the possible conditions for an Authorization policy. Some conditions can be set up so that the current values in the request are compared against static values (A to B), or you can compare static values to current values in the request (B to A). Within one policy, you must probably decide which direction to set up the comparisons and remain consistent unless there is a compelling reason to switch the direction for a particular condition.

For example, suppose you set up a rule to allow access to a resource only during the weekdays (Monday through Friday). You set up four of these conditions to compare if the date when the request is made matches with Monday, Tuesday, Wednesday, or Thursday. You set up the fifth condition to compare whether Friday matches the date when the request is made. This works, but maintaining this policy is more difficult because each new policy manager will look at the Friday condition and wonder why it is configured differently.

Many conditions, when used as the sole condition of a rule, do not make very useful rules. For example, you can create a rule that grants access if the user specifies a specific URL in the request. Such a rule has limited application. But a rule that requires that the request contain a specific URL and that the user have a specific role has greater application because it can be used to limit access to the URL based on the user's role. For information about how conditions can be ANDed or ORed together or placed in different condition groups, see "Using Multiple Conditions" on page 645.

Authorization policies use the following conditions:

- "Authentication Contract Condition" on page 663
- "Client IP Condition" on page 665
- "Credential Profile Condition" on page 666
- "Current Date Condition" on page 667
- "Day of Week Condition" on page 668
- "Current Day of Month Condition" on page 670
- "Current Time of Day Condition" on page 670
- "HTTP Request Method Condition" on page 672
- "LDAP Attribute Condition" on page 673
- "LDAP OU Condition" on page 676
- "Liberty User Profile Condition" on page 677
- "Roles Condition" on page 678
- "Risk Score" on page 679
- "OAuth Scopes" on page 679
- "URL Condition" on page 680
- "URL Scheme Condition" on page 681
- "URL Host Condition" on page 682
- "URL Path Condition" on page 683
- "URL File Name Condition" on page 685
- "URL File Extension Condition" on page 686
- "Virtual Attribute Condition" on page 687
- "X-Forwarded-For IP Condition" on page 688
- "Condition Extension" on page 689
- "Data Extension" on page 690

- “Using the URL Dredge Option” on page 690
- “Edit Button” on page 690

For the specific policies they can be used in “Creating Access Gateway Authorization Policies” on page 653

Authentication Contract Condition

The Authentication Contract condition matches the contract the user logged in with to the contract specified in this condition. The Identity Server has the following default contracts:

Name	URI
Name/Password - Basic	basic/name/password/uri
Name/Password - Form	name/password/uri
Secure Name/Password - Basic	secure/basic/name/password/uri
Secure Name/Password - Form	secure/name/password/uri

To configure other contracts for your system, click **Devices** > **Identity Servers** > **Edit** > **Local** > **Contracts**.

To specify an Authentication Contract condition, fill in the following fields:

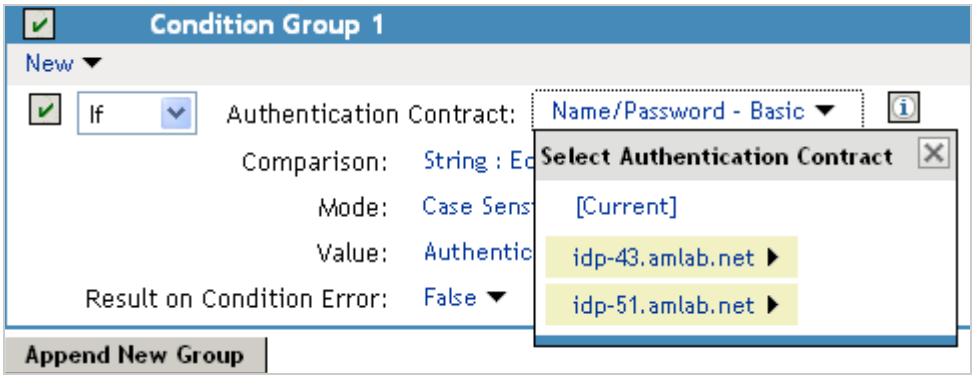
Authentication Contract: To compare the contract that the user used with a static value, select **Current**. To compare a static value with what the user used, select a contract from the list.

If you have created more than one Identity Server configuration, select the configuration that corresponds to the configuration your Access Gateway is configured to trust, then select the contract. The name of the contract is displayed. When you select this name, the configurations that contain a definition for this contract are highlighted.

If you select a contract that is defined on only one of your configurations, be aware that you must change this policy when you change configurations. If you select a contract that is defined in all your configurations, this policy requires no modifications and continues to function when you change configurations.

For example, the following policy has selected Name/Password - Basic as the contract:

Figure 6-25 *An Authentication Contract Defined by Multiple Identity Server Configurations*



Two Identity Server configurations have been defined (idp-43.amlab.net and idp-51.amlab.net). Both configurations are highlighted because Name/Password - Basic is a contract that is automatically defined for all Identity Server configurations. Because it is defined on both configurations, this policy's function is the same, regardless of which configuration is selected as the trusted configuration.

Comparison: Specify how the contract is compared to the data in the **Value** field. Select either a string comparison or a regular expression:

- **Comparison: String:** Specifies that you want the values compared as strings and how you want the string values compared. Select one of the following:
  - **Equals:** Indicates that the values must match, letter for letter.
  - **Starts with:** Indicates that the Authentication Contract value must begin with the letters specified in the **Value** field.
  - **Ends with:** Indicates that the Authentication Contract value must end with the letters specified in the **Value** field.
  - **Contains Substring:** Indicates that the Authentication Contract value must contain the letters, in the same sequence, as specified in the **Value** field.
- **Comparison: Regular Expression: Matches:** Specifies that you want the values compared as regular expressions.

Mode: Select the mode appropriate for the comparison type:

- **Comparison: String:** Specify whether case is important by selecting **Case Sensitive** or **Case Insensitive**.
- **Comparison: Regular Expression: Matches:** Select one or more of the following:

  Canonical Equivalence
  Case Insensitive
  Comments
  Dot All
  Multi-Line
  Unicode
  Unix Lines

  For regular expression syntax information, see the Javadoc for java.util.regex.Pattern.

Value: Specify the value you want to compare with the Authentication Contract value. If you select a static value for the Authentication Contract value, select **Authentication Contract** and **Current**. If you select **Current** for the Authentication Contract value, select **Authentication Contract**, then select the name of a contract.

Other value types are possible if you selected **Current** for the Authentication Contract value. For example:

- You can select **Data Entry Field**. The value specified in the text box must be the URI of the contract for the conditions to match. For a list of these values, click **Access Manager** > **Identity Servers** > **Edit** > **Local** > **Contracts**.
- If you have defined a Liberty User Profile attribute for the URI of authentication contracts, you can select **Liberty User Profile** and your defined attribute.
- If you have defined an LDAP attribute for the URI of the authentication contracts, you can select **LDAP Attribute** and your defined attribute.

Result on Condition Error: Specify what the condition returns when the comparison of the two values returns an error rather than the results of the comparison. Select either **False** or **True**. If you do not want the action applied when an error occurs, select **False**. If you want the action applied when an error occurs, select **True**.

Client IP Condition

The Client IP condition allows you to use the IP address of the user making the request to determine whether the user is allowed access to a resource.

NOTE: Client IP will support IPv4 addresses and not IPv6 addresses.

Fill in the following fields:

Comparison: Specify how the client IP address is compared to the data in the **Value** field. Select either an IP comparison or a regular expression:

- **Comparison: IP:** Specifies that you want the values compared as IP addresses. Select one of the following:
  - **Equals:** Allows you to specify an IP address that the client must match. You can specify more than one.
  - **In Range:** Allows you to specify a range of IP addresses that the client's address must fall within. You can specify more than one range.
  - **In Subnet:** Allows you to specify the subnet that the client's address must belong to. You can specify more than one subnet.
- **Comparison: Regular Expression: Matches:** Specifies that you want the values compared as regular expressions. If you select this option, you must also specify a mode. Select one or more of the following:

  Canonical Equivalence
  Case Insensitive
  Comments
  Dot All
  Multi-Line
  Unicode
  Unix Lines

  For regular expression syntax information, see the Javadoc for java.util.regex.Pattern.

Value: Select **Data Entry Field** and specify a value appropriate for your comparison type. Use the **Edit** button to access a text box where you can enter multiple values, each on a separate line. (For more information, see "Edit Button" on page 690.) Use the **Add** button to add values one at a time. For example:

Comparison Type	Value
Equals	`10.10.10.10` `10.10.10.11`
In Range	`10.10.10.10 - 10.10.10.100` `10.10.20.10 - 10.10.20.100`
In Subnet	`10.10.10.12 / 22` `10.10.20.30 / 22`

Other values types are possible. For example, if your user store contains an LDAP attribute with the IP address of your users, you could select to compare the client's current IP address with the stored value by using an LDAP attribute or a Liberty User Profile value.

Result on Condition Error: Specify what the condition returns when the comparison of the two values returns an error rather than the results of the comparison. Select either **False** or **True**. If you do not want the action applied when an error occurs, select **False**. If you want the action applied when an error occurs, select **True**.

Credential Profile Condition

The Credential Profile condition allows you to control access based on the credentials the user entered when authenticating to the system.

To set up the matching for this condition, fill in the following fields:

Credential Profile: Specify the type of credential your users are using for authentication. If you have created a custom contract that uses credentials other than the ones listed below, do not use the Credential Profile as a condition.

To configure the Credential Profile condition, select one of the following:

- **LDAP Credentials:** If you prompt the user for a username, select this option, then select **LDAP User Name** (the cn of the user), **LDAP User DN** (the fully distinguished name of the user), or **LDAP Password**.

  The default contracts assign the cn attribute to the Credential Profile. If your user store is an Active Directory server, the SAMAccountName attribute is used for the username and stored in the cn field of the LDAP Credential Profile.
- **X509 Credentials:** If you prompt the user for a certificate, select this option, then select one of the following:
  - **X509 Public Certificate Subject:** Retrieves the subject field from the certificate, which can match the DN of the user, depending upon who issued the certificate.
  - **X509 Public Certificate Issuer:** Retrieves the issuer field from the certificate, which is the name of the certificate authority (CA) that issued the certificate.
  - **X509 Public Certificate:** Retrieves the entire certificate, Base64 encoded.
  - **X509 Serial Number:** Retrieves the serial number of the certificate.
- **SAML Credential:** If your users authenticate using a SAML assertion, select this option.

Comparison: Select one of the following types:

- **Comparison: String:** Specifies that you want the values compared as strings and how you want the string values compared. Select one of the following:
  - **Equals:** Indicates that the values must match, letter for letter.
  - **Starts with:** Indicates that the Credential Profile value must begin with the letters specified in the **Value** field.
  - **Ends with:** Indicates that the Credential Profile value must end with the letters specified in the **Value** field.
  - **Contains Substring:** Indicates that the Credential Profile value must contain the letters, in the same sequence, as specified in the **Value** field.
- **Comparison: Regular Expression: Matches:** Specifies that you want the values compared as regular expressions.

Mode: Select the mode appropriate for the comparison type:

- **Comparison: String:** Specify whether case is important by selecting **Case Sensitive** or **Case Insensitive**.
- **Comparison: Regular Expression: Matches:** Select one or more of the following:

  Canonical Equivalence
  Case Insensitive
  Comments
  Dot All
  Multi-Line
  Unicode
  Unix Lines

  For regular expression syntax information, see the Javadoc for java.util.regex.Pattern.

Value: Specify the second value for the comparison. Select one of the following data types:

- **LDAP Attribute:** If you have an LDAP attribute that corresponds to the Credential Profile you have specified, select this option and the attribute.
- **Liberty User Profile:** If you have a Liberty User Profile attribute that corresponds to the Credential Profile you have specified, select this option and the attribute.
- **Data Entry Field:** Specify the string you want matched. Be aware of the following requirements:
  - If you selected **LDAP User DN** as the credential, you need to specify the DN of the user in the **Value** text box. If the comparison type is set to **Contains Substring**, you can match a group of users by specifying a common object that is part of their DNs, for example ou=sales.
  - If you selected **X509 Public Certificate Subject** as the credential, you need to specify all elements of the Subject Name of the certificate in the **Value** text box. Separate the elements with a comma and a space, for example, o=novell, ou=sales. If the comparison type is set to **Contains Substring**, you can match a group of certificates by specifying a name that is part of their Subject Name, for example ou=sales.

Other values are possible. Your policy requirements determine whether they are useful.

Result on Condition Error: Specify what the condition returns when the comparison of the two values returns an error rather than the results of the comparison. Select either **False** or **True**. If you do not want the action applied when an error occurs, select **False**. If you want the action applied when an error occurs, select **True**.

Current Date Condition

The Current Date condition allows you to use the date to determine whether the user is allowed access to a resource.

Fill in the following fields:

Comparison: Specify how the current date is compared to the data in the **Value** field. Select one of the following types:

- **Comparison: Date:** Specifies that you want the values compared as dates. Select one of the following date operators:
  - **Equals:** Requires that the current date must equal the specified value.
  - **Greater Than:** Requires that the current date be after the specified value.

- **Greater Than or Equal to:** Requires that the current date be after or equal to the specified value.
- **Less Than:** Requires that the current date be before the specified value.
- **Less Than or Equal to:** Requires that the current date be before or equal to the specified value.

- **Comparison: Regular Expression: Matches:** Specifies that you want the values compared as regular expressions. Be aware the regular expression matching uses the entire date of the server in its matching. Therefore if the value you are matching is 8, the 8 can produce a match for the year (2008), the month (8), and the day (8, 18, 28).

  If you select this option, you must also specify a mode. Select one or more of the following:

  Canonical Equivalence
  Case Insensitive
  Comments
  Dot All
  Multi-Line
  Unicode
  Unix Lines

  For regular expression syntax information, see the Javadoc for java.util.regex.Pattern.

Date Format: If you selected a date comparison, specify the format of the **Value** field. Select one of the following formats:

- **D/M/Y =** 1/Jul/2009 or 1/7/2009
- **D-M-Y =** 1-Jul-2009 or 1-7-2009
- **D.M.Y =** 1.Jul.2009 or 1.7.2009
- **M/D/Y =** Jul/1/2009 or 7/1/2009
- **M-D-Y =** Jul-1-2009 or 7-1-2009
- **M.D.Y =** Jul.1.2009 or 7.1.2009
- **YYYY-MM-DD =** 2009-07-01
- **YYYY.MM.DD =** 2009.07.01

D specifies a number from 1 to 31. **M** specifies a number from 1 to 12 or the name of the month in three letters (Sep) or complete (September). **Y** specifies the year in a four-digit format.

Value: Specify the second value for the comparison. If you select **Data Entry Field** as the value type, specify the date in the format you select in the **Date Format** field.

Other value types are possible. Your policy requirements determine whether they are useful. If you have set up a Liberty User Profile or an LDAP attribute that corresponds to the date, you can use this option and select your attribute. The **Date Format** field does not apply to these value types.

Result on Condition Error: Specify what the condition returns when the comparison of the two values returns an error rather than the results of the comparison. Select either **False** or **True**. If you do not want the action applied when an error occurs, select **False**. If you want the action applied when an error occurs, select **True**.

Day of Week Condition

The Current Day of Week condition allows you to restrict access based on which day of the week the request is made. Fill in the following fields:

Current Day of Week: Select the name of the day from the list. To compare the day specified in the current request with a static value, select **Current**. To compare a static value with the day specified in the current request, select the name of a day from the list.

Comparison: Specify how the current day of the week is compared to the data in the **Value** field. Select one of the following types:

- **Comparison: Day of Week:** Specifies that you want the values compared as a day of the week. Select one of the following operators:
  - **Equals:** Allows you to specify a day that the client must match.
  - **In Range:** Allows you to specify a range of days that the client's request must fall within, for example, Monday to Friday.
- **Comparison: Regular Expression: Matches:** Specifies that you want the values compared as regular expressions. Be aware that regular expression matching uses the entire date of the server in its matching. Therefore if the value you are matching is M, the M can produce a match for months (March and May) and for time zones (such as MST).

  If you select this option, you must also specify a mode. Select one or more of the following:

  Canonical Equivalence
  Case Insensitive
  Comments
  Dot All
  Multi-Line
  Unicode
  Unix Lines

  For regular expression syntax information, see the Javadoc for java.util.regex.Pattern.

Value: Specify the second value for the comparison. If you select **Current** for the **Current Day of Week** field, you need to specify a static value. If you select a static value for the **Current Day of the Week** field, you need to select **Current** for the **Value** field.

If you select **Data Entry Field** as the value type, days of the week are specified in the following format:

```
Sun or Sunday
Mon or Monday
Tue or Tuesday
Wed or Wednesday
Thu or Thursday
Fri or Friday
Sat or Saturday
```

If you selected **In Range** as the comparison type, specify the first day of the range in the left text box and the end day of the range in the right text box.

Other value types are possible. Your policy requirements determine whether they are useful. If you have set up a Liberty User Profile or an LDAP attribute that corresponds to a day of the week, you can use this option and select your attribute.

Result on Condition Error: Specify what the condition returns when the comparison of the two values returns an error rather than the results of the comparison. Select either **False** or **True**. If you do not want the action applied when an error occurs, select **False**. If you want the action applied when an error occurs, select **True**.

Current Day of Month Condition

The Current Day of Month condition allows you to restrict access based on the day of the month the request is made. Fill in the following fields:

Comparison: Specify how the current day of the month is compared to the data in the **Value** field. Select one of the following types:

- **Comparison: Day of Month:** Specifies that you want the values compared as a day of the month. Select one of the following operators:
  - **Equals:** Allows you to specify a day that the client must match.
  - **In Range:** Allows you to specify a range of days that the client's request must fall within.
- **Comparison: Regular Expression: Matches:** Specifies that you want the values compared as regular expressions. Regular expression matching uses the entire date of the server in its matching. Therefore if the value you are matching is 8, the 8 can produce a match for the year (2008), the month (8), and the day (8, 18, 28). If you want to match only on a day of the month (1-31), you need to use the Day of Month comparison rather than a Regular Expression comparison.

  If you select this option, you must also specify a mode. Select one or more of the following:

  Canonical Equivalence
  Case Insensitive
  Comments
  Dot All
  Multi-Line
  Unicode
  Unix Lines

  For regular expression syntax information, see the Javadoc for java.util.regex.Pattern.

Value: Specify the second value for the comparison:

- If you select **Equals** for the comparison type, you would normally select **Data Entry Field** for the **Value** field and specify a number from 1 to 31 in the text box.
- If you select **In Range** for the comparison type, you would normally select **Data Entry Field** for the **Value** field and specify the first value of the range in the first text box and the second value of the range in the second text box. If you specify 1 in the first box and 15 in the second box, you can use this condition to restrict access between the first day of the month and the 15th day.

Other value types are possible. Your policy requirements determine whether they are useful. If you have set up a Liberty User Profile or an LDAP attribute that corresponds to a day of the month, you can use this option and select your attribute.

Result on Condition Error: Specify what the condition returns when the comparison of the two values returns an error rather than the results of the comparison. Select either **False** or **True**. If you do not want the action applied when an error occurs, select **False**. If you want the action applied when an error occurs, select **True**.

Current Time of Day Condition

The Current Time of Day condition allows you to restrict access based on the time the request is made. Fill in the following fields:

Comparison: Specify how the current time of day is compared to the data in the **Value** field. Select one of the following types:

- **Comparison: Time:** Specifies that you want the values compared as time. Select one of the following:
  - **Greater Than:** Requires that the current time is greater than the specified value.
  - **Greater Than or Equal to:** Requires that the current time is greater than or equal to the specified value.
  - **Less Than:** Requires that the current time is less than the specified value.
  - **Less Than or Equal to:** Requires that the current time is less than or equal to the specified value.
  - **In Range:** Requires that the current time must fall within the specified range, such as 08:00 and 17:00.

  If you specify this type of comparison, you must also specify a time zone. Select either the **Local** time zone or **GMT** (Greenwich Mean Time).
- **Comparison: Regular Expression: Matches:** Specifies that you want the values compared as regular expressions. Regular expression matching uses the entire date and time of the server in its matching. Therefore if the value you are matching is 8, the 8 can produce a match for the year (2008), the month (8), the day (8, 18, 28), the hour (8), the minute (8, 18, 28, 38, 48) and the second (8, 18, 28, 38, 48).

  If you select this option, you must also specify a mode. Select one or more of the following:

  Canonical Equivalence
  Case Insensitive
  Comments
  Dot All
  Multi-Line
  Unicode
  Unix Lines

  For regular expression syntax information, see the Javadoc for java.util.regex.Pattern.

Value: Specify the second value for the comparison. If you select **Data Entry Field** as the value type, hours and minutes are specified in the following format:

```
hour:minute
```

Hour is a number from 00 to 23, and minute is a number from 00 to 59.

Time can only be specified in a 24-hour clock format. For example, 8 am is 08:00 and 5:30 pm is 17:30.

Other value types are possible. Your policy requirements determine whether they are useful. If you have set up a Liberty User Profile or an LDAP attribute that corresponds to the time of day, you can use this option and select your attribute.

Result on Condition Error: Specify what the condition returns when the comparison of the two values returns an error rather than the results of the comparison. Select either **False** or **True**. If you do not want the action applied when an error occurs, select **False**. If you want the action applied when an error occurs, select **True**.

HTTP Request Method Condition

The HTTP Request Method condition allows you to restrict accessed based on the request method in the current request.

HTTP Request Method: Select the request method from the list or select **Current** to specify the method in the current request.

Comparison: Specify how the HTTP Request Method is compared to the data in the **Value** field. Select one of the following types:

- **Comparison: String:** Specifies that you want the values compared as strings and how you want the string values compared. Select one of the following:
  - **Equals:** Indicates that the values must match, letter for letter.
  - **Starts with:** Indicates that the HTTP Request Method value must begin with the letters specified in the **Value** field.
  - **Ends with:** Indicates that the HTTP Request Method value must end with the letters specified in the **Value** field.
  - **Contains Substring:** Indicates that the HTTP Request Method value must contain the letters, in the same sequence, as specified in the **Value** field.
- **Comparison: Regular Expression: Matches:** Specifies that you want the values compared as regular expressions.

Mode: Select the mode appropriate for the comparison type:

- **Comparison: String:** Specify whether case is important by selecting **Case Sensitive** or **Case Insensitive**.
- **Comparison: Regular Expression: Matches:** Select one or more of the following:

  Canonical Equivalence
  Case Insensitive
  Comments
  Dot All
  Multi-Line
  Unicode
  Unix Lines

  For regular expression syntax information, see the Javadoc for java.util.regex.Pattern.

Value: Specify the value you want compared to the HTTP Request Method value. If you selected a method from the list for the HTTP Request Method value, select **HTTP Request Method** > **Current**. If you selected **Current** for the HTTP Request Method value, select a request method from the list.

Other value types are possible. Your policy requirements determine whether they are useful. If you have set up a Liberty User Profile or an LDAP attribute that corresponds to an HTTP Request Method, you can use this option and select your attribute.

Result on Condition Error: Specify what the condition returns when the comparison of the two values returns an error rather than the results of the comparison. Select either **False** or **True**. If you do not want the action applied when an error occurs, select **False**. If you want the action applied when an error occurs, select **True**.

LDAP Attribute Condition

The LDAP Attribute condition allows you to restrict access based on a value in an LDAP attribute defined for the inetOrgPerson class or any other LDAP attribute you have added. You can have the user's attribute value retrieved from your LDAP directory and compared to a value of the following type:

- Roles from an identity provider
- Date and time and its various elements
- URL and its various elements
- IP address
- Authentication contract
- Credential profile
- HTTP request method
- Liberty User Profile attribute
- Static value in a data entry field

This condition is one of the slower conditions to process because the value needs to be retrieved from the LDAP server. If the value is not time sensitive, you can have attribute value sent in the assertion when the user authenticates. Its value is then in cache and available. For configuration information, click **Devices > Identity Servers > Servers > Edit > Liberty [or SAML 1.0 or SAML 2.0] > [Provider] > Attributes**.

To set up the matching for this condition, fill in the following fields:

LDAP Attribute: Specify the LDAP attribute you want to use in the comparison. Select from the listed LDAP attributes. To add an attribute that isn't in the list, scroll to the bottom of the list, click **New LDAP Attribute**, then specify the name of the attribute.

Refresh Data Every: Sends a query to the LDAP server to verify the current value of the attribute according to the specified interval. Because querying the LDAP server slows down the processing of a policy, LDAP attribute values are normally cached after the value has been obtained. The default cache interval is for the user session. You must change the value of this option from Session to a more frequent interval only on those attributes that are critical to the security of your system or to the design of your work flow.

You can select to cache the value for the session, for the request, or for a time interval varying from 5 seconds to 60 minutes. For more information about this option, see "Using the Refresh Data Option" on page 651.

Comparison: Specify how you want the values compared. All data types are available. Select one of the following that matches the value type of your attribute:

- **Date:** Specifies that you want the values compared as dates. Select one of the following date operators:
  - **Equals:** Indicates that the current date must be equal to the specified value.
  - **Greater Than:** Indicates that the current date be after the specified value.
  - **Greater Than or Equal to:** Indicates that the current date be after or equal to the specified value.
  - **Less Than:** Indicates that the current date be before the specified value.
  - **Less Than or Equal to:** Indicates that the current date be before or equal to the specified value.

- **Day of Week:** Specifies that you want the values compared as a day of the week. Select one of the following operators:
  - **Equals:** Allows you to specify a day that the specified value must match.
  - **In Range:** Allows you to specify a range of days that the specified value must fall within, for example, Monday to Friday.
- **Day of Month:** Specifies that you want the values compared as a day of the month. Select one of the following operators:
  - **Equals:** Allows you to specify a day that the specified value must match.
  - **In Range:** Allows you to specify a range of days that the specified value must fall within.
- **Integer:** Specifies that you want the values compared as integers. Select one of the following:
  - **Equals:** Indicates that the integer value must be equal to the specified value.
  - **Greater Than:** Indicates that the integer value must be greater than the specified value.
  - **Greater Than or Equal to:** Indicates that the integer value must be greater than or equal to the specified value.
  - **Less Than:** Indicates that the integer value is less than the specified value.
  - **Less Than or Equal to:** Indicates that the integer value is less than or equal to the specified value.
- **IP:** Specifies that you want the values compared as IP addresses. Select one of the following:
  - **Equals:** Allows you to specify an IP address that the specified value must match. You can specify more than one.
  - **In Range:** Allows you to specify a range of IP addresses that the specified value must fall within. You can specify more than one range.
  - **In Subnet:** Allows you to specify the subnet that the specified value must belong to. You can specify more than one subnet.
- **LDAP OU: Contains:** Specifies that you want the condition to determine whether the user is contained by a specified organizational unit.
- **Attribute: Does Exist?** Specifies that you want the condition to determine whether the user has an LDAP attribute. This is a unary condition.
- **Regular Expression: Matches:** Specifies that you want the values compared as regular expressions.
- **String:** Specifies that you want the values compared as strings and how you want the string values to be compared. Select one of the following:
  - **Equals:** Indicates that the values must match, letter for letter.
  - **Starts with:** Indicates that the attribute value must begin with the letters specified in the **Value** field.
  - **Ends with:** Indicates that the attribute value must end with the letters specified in the **Value** field.
  - **Contains Substring:** Indicates that the attribute value must contain the letters, in the same sequence, as specified in the **Value** field.
- **Time:** Specifies that you want the values compared as time. Select one of the following:
  - **Greater Than:** Indicates that the current time is greater than the specified value.
  - **Greater Than or Equal to:** Indicates that the current time is greater than or equal to the specified value.
  - **Less Than:** Indicates that the current time is less than the specified value.

- **Less Than or Equal to:** Indicates that the current time is less than or equal to the specified value.
- **In Range:** Indicates that the current time must fall within the specified range, such as 08:00 and 17:00.

- **URL: Equals:** Specifies that you want the values compared as URLs.
- **URL Scheme:** Specifies that you want the values compared as scheme strings and how you want the values compared. Select one of the following:
  - **Equals:** Indicates that the URL scheme must contain the same letters, in the same order as specified in the value.
  - **Starts with:** Indicates that the URL scheme must begin with the letters specified in the value.
  - **Ends with:** Indicates that the URL scheme must end with the letters specified in the value.
  - **Contains Substring:** Indicates that the URL scheme must contain the letters, in the same sequence, as specified in the value.
- **URL Host: Equals:** Specifies that you want the values compared as hostnames.
- **URL Path:** Specifies that you want the values compared as paths and how you want the string values compared. Select one of the following:
  - **Equals:** Indicates that the URL path must contain the same letters, in the same order as specified in the value.
  - **Starts with:** Indicates that the URL path must begin with the letters specified in the value.
  - **Ends with:** Indicates that the URL path must end with the letters specified in the value.
  - **Contains SUbstring:** Indicates that the URL path must contain the letters, in the same sequence, as specified in the Value field.
- **URL File:** Specifies that you want the values compared as filenames and how you want the names compared. Select one of the following:
  - **Equals:** Indicates that the filenames must contain the same letters, in the same order as specified in the value.
  - **Starts with:** Indicates that the filenames must begin with the letters specified in the value.
  - **Ends with:** Indicates that the filenames must end with the letters specified in the value.
  - **Contains Substring:** Indicates that the filenames must contain the letters, in the same sequence, as specified in the Value field.
- **URL File Extension:** Specifies that you want the values compared as file extensions and how you want the file extensions compared. Select one of the following:
  - **Equals:** Indicates that the file extensions must contain the same letters, in the same order as specified in the value.
  - **Starts with:** Indicates that the file extensions must begin with the letters specified in the value.
  - **Ends with:** Indicates that the file extensions must end with the letters specified in the value.
  - **Contains Substring:** Indicates that the file extensions must contain the letters, in the same sequence, as specified in the Value field.

Mode: Select the mode, if available, that matches the comparison type. For example, if you select to compare the values as strings, you can select either a **Case Sensitive** mode or a **Case Insensitive** mode.

Value: Specify the second value for the comparison. All data types are available. For example, you can select to compare the value of one LDAP attribute to the value of another LDAP attribute. Only you can determine if such a comparison is meaningful.

Result on Condition Error: Specify what the condition returns when the comparison of the two values returns an error rather than the results of the comparison. Select either **False** or **True**. If you do not want the action applied when an error occurs, select **False**. If you want the action applied when an error occurs, select **True**.

LDAP OU Condition

The LDAP OU condition allows you to compare the DN of an OU against the DN that was used when the user authenticated. If the user's DN contains the OU, the condition matches.

LDAP OU: Select **[Current]**.

Comparison: Specify how you want the values compared. Select one of the following:

- **Contains:** Specifies that you want the condition to determine whether the user is contained by a specified organizational unit.
- **Comparison: Regular Expression: Matches:** Specifies that you want the values compared as regular expressions.

Mode: Select the mode appropriate for the comparison type.

- **Contains:** Select whether the user must be contained in the specified OU (**One Level)** or whether the user can be contained in the specified OU or a child container (**Subtree)**.
- **Comparison: Regular Expression: Matches:** Select one or more of the following:

  Canonical Equivalence
  Case Insensitive
  Comments
  Dot All
  Multi-Line
  Unicode
  Unix Lines

  For regular expression syntax information, see the Javadoc for java.util.regex.Pattern.

Value: Specify the second value for the comparison. If you select **LDAP OU** > **Name of Identity Server Configuration** > **User Store Name**, you can browse to the name of the OU.

If you have more than 250 OUs defined in your tree, you are prompted to enter an LDAP query string. In the text box, you need to add only the <strFilter> value for the query. For example:

<strFilter> Value	Description
admin*	Returns all OUs that begin with admin, such as adminPR, adminBG, and adminWTH.
*test	Returns all OUs that end with test, such as doctest, softtest, and securtest.
low	Returns all OUs that have "low" in the name, such as low, yellow, and clowns.

For more information about the <strFilter> parameter, see RFC 2254 "LDAP Search Filter."

If you select **Data Entry Field**, you can enter the DN of the OU in the text field. For example:

```
cn=users,dc=bcf2,dc=provo,dc=novell,dc=com
```

```
ou=users,o=novell
```

If you have defined a Liberty User Profile or an LDAP attribute for the OU you want to match, select this option, then select your attribute.

Result on Condition Error: Specify what the condition returns when the comparison of the two values returns an error rather than the results of the comparison. Select either **False** or **True**. If you do not want the action applied when an error occurs, select **False**. If you want the action applied when an error occurs, select **True**.

Liberty User Profile Condition

The Liberty User Profile condition allows you to restrict access based on a value in a Liberty User Profile attribute. The Liberty attributes must be enabled before you can use them in policies (click **Devices** > **Identity Servers** > **Edit** > **Liberty** > **Web Server Provider**, then enable one or more of the following: **Employee Profile**, **Personal Profile**).

These attributes can be mapped to LDAP attributes (click **Devices** > **Identity Servers** > **Edit** > **Liberty** > **LDAP Attribute Mapping**). When mapped, the actual value comes from your user store. If you are using multiple user stores with different LDAP schemas, mapping similar attributes to the same Liberty User Profile attribute allows you to create one policy with the Liberty User Profile attribute rather than multiple policies for each LDAP attribute.

The selected attribute is compared to a value of the following type:

- Roles from an identity provider
- Date and time and its various elements
- URL and its various elements
- IP address
- Authentication contract
- Credential profile
- HTTP request method
- LDAP attribute
- Static value in a data entry field

To set up the matching for this condition, fill in the following fields:

Liberty User Profile: Select the Liberty User Profile attribute. These attributes are organized into two main groups: Corporate Employment Identity and Entire Personal Identity. By default, the Common Last Name attribute for Liberty User Profile is mapped to the sn attribute for LDAP. To select this attribute for comparison, click **Entire Personal Identity** > **Entire Common Name** > **Common Analyzed Name** > **Common Last Name**.

Comparison: Select the comparison type that matches the data type of the selected attribute and the value.

Mode: Select the mode, if available, that matches the data type. For example, if you select to compare the values as strings, you can select either a **Case Sensitive** mode or a **Case Insensitive** mode.

Value: Select one of the values that is available from the current request or select **Data Entry Field** to enter a static value. The static value that you can enter is dependent upon the comparison type you selected.

Result on Condition Error: Specify what the condition returns when the comparison of the two values returns an error rather than the results of the comparison. Select either **False** or **True**. If you do not want the action applied when an error occurs, select **False**. If you want the action applied when an error occurs, select **True**.

Roles Condition

If you have configured some Access Manager Appliance role policies (see "Creating Roles" on page 608), you can use these roles as conditions to control access. Roles are not assigned to users until the users authenticate. All authenticated users are assigned the `authenticated` role. If you use a comparison type of starts with, ends with, or contains substring, carefully evaluate the potential results. For example, if you specify `ed` as the value for an ends with comparison, the condition matches roles such as `contracted` and `assigned` that you created, but it also matches the `authenticated` role.

Fill in the following fields:

Roles: Select the role. To compare the roles the user is currently assigned with a specific role, select [Current].

Comparison: Select one of the following types:

- **Comparison: String:** Specifies that you want the values compared as strings, and how you want the string values compared. Select one of the following:
  - **Equals:** Indicates that the values must match, letter for letter.
  - **Starts with:** Indicates that the Roles value must begin with the letters specified in the **Value** field.
  - **Ends with:** Indicates that the Roles value must end with the letters specified in the **Value** field.
  - **Contains Substring:** Indicates that the Roles value must contain the letters, in the same sequence, as specified in the **Value** field.
- **Comparison: Regular Expression: Matches:** Specifies that you want the values compared as regular expressions.

Mode: Select the mode appropriate for the comparison type:

- **Comparison: String:** Specify whether case is important by selecting **Case Sensitive** or **Case Insensitive**.
- **Comparison: Regular Expression: Matches:** Select one or more of the following:

  Canonical Equivalence
  Case Insensitive
  Comments
  Dot All
  Multi-Line
  Unicode
  Unix Lines

  For regular expression syntax information, see the Javadoc for java.util.regex.Pattern.

Value: If you have created Identity Server roles policies, select **Roles**, then select the role you want the user to have to match this condition. The `authenticated` role is assigned to all users when they authenticate. If you have defined a Liberty User Profile or an LDAP attribute for a role, you can select this option, then select your attribute.

You can use the **Data Entry Field** option to enter the name of the role you want to test for. If you have activated roles from an external source, use this option to specify the name of the role.

Result on Condition Error: Specify what the condition returns when the comparison of the two values returns an error rather than the results of the comparison. Select either **False** or **True**. If you do not want the action applied when an error occurs, select **False**. If you want the action applied when an error occurs, select **True**.

Risk Score

You can define a condition group as part of the authorization policy that uses the risk score from Identity Server to protect a resource.

1. Specify how the current risk score is compared to the data in the Value field. You can select to do the comparison as an integer value or as a regular expression. For more details about regular expression, see “Comparison: Regular Expression: Matches:” on page 669.
2. Specify the value as a Data Entry and enter the risk score for the rule.
3. Specify the value to return if the rule execution results in an error. Select either **False** or **True**.
4. Configure the actions required to be performed during evaluation of the condition group. For more information, see Step 7 on page 654.
5. Click **OK**.

OAuth Scopes

You can define a condition group as part of the authorization policy that uses the OAuth scopes from the Identity Server to protect a resource.

1. **Comparison:** You can select to do the comparison as an integer value or as a regular expression. For more information about regular expression, see “Comparison: Regular Expression: Matches:” on page 669.
2. **Mode:** Select the mode appropriate for the comparison type. For more information, see “Mode:” on page 667.
3. **Value:** Select `OAuth Scopes`.
4. **Result on Condition Error:** Specify what the condition returns when the comparison of the two values returns an error rather than the results of the comparison. Select either `False` or `True`. If you do not want the action applied when an error occurs, select `False`. If you want the action applied when an error occurs, select `True`.
5. Configure the actions required to be performed during evaluation of the condition group. For more information, see Step 7 on page 654.
6. Click **OK**.

URL Condition

The URL condition allows you to restrict access based on the URL specified in the request. If you have users requesting a resource with a URL you don't want them to use, you can use this condition in an Access Gateway Authorization policy to deny them access to this URL, and use the Actions section to redirect the request to the URL you want them to use.

To set up matching for this condition, fill in the following fields:

Comparison: Specify how the URL is compared to the data in the **Value** field. Select one of the following types:

- **Comparison: URL: Equals:** Specifies that you want the values compared as URLs.
- **Comparison: Regular Expression: Matches:** Specifies that you want the values compared as regular expressions.

Mode: Select the mode appropriate for the comparison type:

- **Comparison: URL: Equals:** Specify whether case is important by selecting **Case Sensitive** or **Case Insensitive**.
- **Comparison: Regular Expression: Matches:** Select one or more of the following:

  Canonical Equivalence
  Case Insensitive
  Comments
  Dot All
  Multi-Line
  Unicode
  Unix Lines

  For regular expression syntax information, see the Javadoc for java.util.regex.Pattern.

  The URL query strings can also be used in comparisons. Select Regular Expression Match to implement it. For example, if you want to match against a query string parameter xyz, whose value is abc, you need to enter the following regular expression in the **Data Entry** field:

  ```
  ".*\?.*xyz=abc.*"
  ```

  This follows the java regular expression pattern. For more information, see *The API specification for the Java 2 Platform* *(https://docs.oracle.com/javase/7/docs/api/java/util/regex/Pattern.html)* for the Java Regular Expression syntax.

Value: To enter a static value to compare to the URL in the current request, select **Data Entry Field** and specify the URL. This must be the complete URL, starting with the URL scheme (http:// or https://) and including the domain name, but not the port. If the URL contains a path, you must include it. If you do not specify a scheme, HTTP is used.

If you selected **Regular Expression: Matches**, regular expression rules apply.

If you selected **URL: Equals** for your comparison type, the wildcard characters (?) or (*) can be specified as the last element of the URL path to aid in matching basic URL patterns. These wildcard characters are interpreted as follows:

- ? matches all files at the specified directory level
- * matches all files and directories at and beyond the specified directory level

For example, if the request URL is `http://www.resourcehost.com/path/resource.gif`, the following entered URLs would match the request URL:

```
http://www.resourcehost.com/path/resource.gif
http://www.resourcehost.com/path/?
http://www.resourcehost.com/path/*
http://www.resourcehost.com/*
```

If you selected **URL:Equals** for the comparison type, you can add multiple values:

- Use the **Edit** button to access a text box where you can enter multiple values, each on a separate line. For more information, see "Edit Button" on page 690.
- Use the **Add** button to add values one at a time.
- Use the **URL Dredge** button to display a list of links to use as values. For more information about this option, see Using the URL Dredge Option.

All entered URLs are compared to the request URL until a match is found or the list is exhausted.

If you have defined a Liberty User Profile or an LDAP attribute for a URL, you can select these options for the value type, then select your attribute.

Result on Condition Error: Specify what the condition returns when the comparison of the two values returns an error rather than the results of the comparison. Select either **False** or **True**. If you do not want the action applied when an error occurs, select **False**. If you want the action applied when an error occurs, select **True**.

URL Scheme Condition

The URL Scheme condition allows you to restrict access based on the scheme specified in the URL of the request. For example in an Access Gateway Authorization policy, if the request contains HTTP as the scheme in the URL and you require users to use HTTPS, you can use this condition to deny access and redirect them to another URL.

This condition allows you to compare A to B or B to A. You need to decide whether you want to compare a static value to the current value in the HTTP request, or whether you want to compare the current value in the HTTP request to a specified value. The comparison type you use depends upon the value you want to specify. If you want more flexibility in specifying the value, you must select to compare the current value in the HTTP request with a specified value.

To set up matching for this condition, fill in the following fields:

URL Scheme: Specify the scheme you want compared. You can select **Current** for the current value in the HTTP request, or specify a static value of **http** or **https**.

Comparison: Select one of the following types:

- **Comparison: URL Scheme:** Specifies that you want the values compared as scheme strings and how you want the values compared. Select one of the following:
  - **Equals:** Indicates that the URL scheme must contain the same letters, in the same order as specified in the value.
  - **Starts with:** Indicates that the URL scheme must begin with the letters specified in the value.
  - **Ends with:** Indicates that the URL scheme must end with the letters specified in the value.
  - **Contains Substring:** Indicates that the URL scheme must contain the letters, in the same sequence, as specified in the **Value** field.
- **Comparison: Regular Expression: Matches:** Specifies that you want the values compared as regular expressions.

Mode: Select the mode appropriate for the comparison type:

- **Comparison: String:** Specify whether case is important by selecting **Case Sensitive** or **Case Insensitive**.
- **Comparison: Regular Expression: Matches:** Select one or more of the following:

  Canonical Equivalence
  Case Insensitive
  Comments
  Dot All
  Multi-Line
  Unicode
  Unix Lines

  For regular expression syntax information, see the Javadoc for java.util.regex.Pattern.

Value: Specify the value you want to compare with the URL Scheme value. If you select a static value for the URL Scheme value, select **URL Scheme** and **Current**. If you select **Current** for the URL Scheme value, select one of the following value types:

- **Data Entry Field:** Allows you to specify the scheme value you want to use in the comparison. The scheme cannot be specified with a trailing colon (:) character and must be specified in lowercase (**http** or **https**). Use the **Edit** button to access a text box where you can enter multiple values, each on a separate line. (For more information, see "Edit Button" on page 690.) Use the **Add** button to add values one at a time.

  All entered URL schemes are compared to the requested URL scheme until a match is found or the list is exhausted.
- **LDAP Attribute:** If you have defined an LDAP attribute containing a URL or URL scheme, you can select this option, then select your attribute.
- **Liberty User Profile:** If you have defined a Liberty User Profile attribute containing a URL or URL scheme, you can select this option, then select your attribute.

Result on Condition Error: Specify what the condition returns when the comparison of the two values returns an error rather than the results of the comparison. Select either **False** or **True**. If you do not want the action applied when an error occurs, select **False**. If you want the action applied when an error occurs, select **True**.

URL Host Condition

The URL Host condition allows you to restrict access based on the hostname specified in the URL of the request. For example, you can use this condition to create rules that allow access if the URL contains one hostname, but deny access if the URL contains another hostname. The URL Host condition compares the hostname in the URL of the current request to the URL hostname specified in the **Value** field.

To set up matching for this condition, fill in the following fields:

Comparison: Specify how the URL Host is compared to the data in the **Value** field. Select one of the following types:

- **Comparison: URL Host: Equals:** Specifies that you want the values compared as hostnames.
- **Comparison: Regular Expression: Matches:** Specifies that you want the values compared as regular expressions. If you select this option, you must also specify a **Mode**. Select one or more of the following:

Canonical Equivalence
Case Insensitive
Comments
Dot All
Multi-Line
Unicode
Unix Lines

For regular expression syntax information, see the Javadoc for java.util.regex.Pattern.

Value: Select one of the following value types, then specify a value:

- **Data Entry Field:** To specify a static value to compare to the URL host in the current request, select this value type and specify the DNS name of the host.

  For example, if the request URL is http://www.resourcehost.com/path/resource.gif, the following hostname matches the resource URL:

  ```
  www.resourcehost.com
  ```

  If you selected **URL Host:Equals** for the comparison type, you can add multiple values:

  - Use the **Edit** button to access a text box where you can enter multiple values, each on a separate line. For more information, see "Edit Button" on page 690.
  - Use the **Add** button to add values one at a time.

  All listed hostnames are compared to the requested URL until a match is found or the list is exhausted.
- **LDAP Attribute:** If you have defined an LDAP attribute containing a URL or URL host, you can select this option, then select your attribute.
- **Liberty User Profile:** If you have defined a Liberty User Profile attribute containing a URL or URL host, you can select this option, then select your attribute.

Result on Condition Error: Specify what the condition returns when the comparison of the two values returns an error rather than the results of the comparison. Select either **False** or **True**. If you do not want the action applied when an error occurs, select **False**. If you want the action applied when an error occurs, select **True**.

URL Path Condition

The URL Path condition allows you to restrict access based on the path specified in the URL of the request. This condition compares the path of the URL in the current request to the path specified in the **Value** field.

To set up matching for this condition, fill in the following fields:

Comparison: Select one of the following types:

- **Comparison: URL Path:** Specifies that you want the values compared as paths and how you want the string values compared. Select one of the following:
  - **Equals:** Indicates that the URL path must contain the same letters, in the same order as specified in the value.
  - **Starts with:** Indicates that the URL path must begin with the letters specified in the value.
  - **Ends with:** Indicates that the URL path must end with the letters specified in the value.

- **Contains Substring:** Indicates that the URL path must contain the letters, in the same sequence, as specified in the **Value** field.

- **Comparison: Regular Expression: Matches:** Specifies that you want the values compared as regular expressions.

Mode: Select the mode appropriate for the comparison type:

- **Comparison: URL Path:** Specify whether case is important by selecting **Case Sensitive** or **Case Insensitive**.
- **Comparison: Regular Expression: Matches:** Select one or more of the following:

  Canonical Equivalence
  Case Insensitive
  Comments
  Dot All
  Multi-Line
  Unicode
  Unix Lines

  For regular expression syntax information, see the Javadoc for java.util.regex.Pattern.

Value: Specify the value type and value for the comparison. Select one of the following:

- **Data Entry Field:** To enter a static value to compare to the URL path in the current request, select this value type and specify the path. Start the path with a forward slash.

  IMPORTANT: If you need to add a space in the path, you need to enter this encoded value for the space: %20

  If you have selected **Regular Expression: Matches** for your comparison type, regular expression rules apply. If you have selected **URL Path** for your comparison type, the path can end with a filename or a wildcard. An asterisk (*) matches all files and directories at and beyond the specified directory level. A question mark (?) matches all files at the specified directory level. For example:

Path	Match Description
`/path1/path2/`	Requires an exact match of the URL path. It matches if the URL does not contain anything after `path2`.
`/path1/file.ext`	Requires an exact match of the URL path, including the extension on the filename.
`/path1/path2/?`	Matches everything that immediately follows `path2`. It does not match anything if the path contains another directory, such as `/path1/path2/path3/file3.ext`.
`/path1/path2/*`	Matches everything that follows `path2`, including a filename or another directory, such as `/path1/path2/path3/file3.ext`.

  If you selected **URL Path** for the comparison type, you can add multiple values:

  - Use the **Edit** button to access a text box where you can enter multiple values, each on a separate line. For more information, see “Edit Button” on page 690.
  - Use the **Add** button to add values one at a time.

All entered URL paths are compared to the request URL path until a match is found or the list is exhausted.

- **LDAP Attribute:** If you have defined an LDAP attribute containing a URL or URL path, you can select this option, then select your attribute.
- **Liberty User Profile:** If you have defined a Liberty User Profile attribute containing a URL or URL path, you can select this option, then select your attribute.

Result on Condition Error: Specify what the condition returns when the comparison of the two values returns an error rather than the results of the comparison. Select either **False** or **True**. If you do not want the action applied when an error occurs, select **False**. If you want the action applied when an error occurs, select **True**.

URL File Name Condition

The URL File Name condition allows you to restrict access based on the filename specified in the URL. It compares the filename in the URL of the current request to the filename specified in the **Value** field.

To set up matching for this condition, fill in the following fields:

Comparison: Select one of the following types:

- **Comparison: URL File:** Specifies that you want the values compared as filenames and how you want the names compared. Select one of the following:
  - **Equals:** Indicates that the filenames must contain the same letters, in the same order as specified in the value.
  - **Starts with:** Indicates that the filenames must begin with the letters specified in the value.
  - **Ends with:** Indicates that the filenames must end with the letters specified in the value.
  - **Contains Substring:** Indicates that the filenames must contain the letters, in the same sequence, as specified in the **Value** field.
- **Comparison: Regular Expression: Matches:** Specifies that you want the values compared as regular expressions.

Mode: Select the mode appropriate for the comparison type:

- **Comparison: URL File:** Specify whether case is important by selecting **Case Sensitive** or **Case Insensitive**.
- **Comparison: Regular Expression: Matches:** Select one or more of the following:

  Canonical Equivalence
  Case Insensitive
  Comments
  Dot All
  Multi-Line
  Unicode
  Unix Lines

  For regular expression syntax information, see the Javadoc for java.util.regex.Pattern.

Value: Specify the value type and value for the comparison. Select one of the following:

- **Data Entry Field:** To specify a static value to compare to the filename in the current request, select this value type and specify the filename.

The value you specify is compared to what follows the last slash in the URL. If you selected **Regular Expression: Matches** for your comparison type, regular expression rules apply. If you selected **URL File** for your comparison type, enter a value that matches your string comparison type. Do not use wildcards in your value.

If you selected **URL File** for the comparison type, you can add multiple values:

- Use the **Edit** button to access a text box where you can enter multiple values, each on a separate line. For more information, see "Edit Button" on page 690.
- Use the **Add** button to add values one at a time.

All listed filenames are compared to the requested URL filename until a match is found or the list is exhausted.

- **LDAP Attribute:** If you have defined an LDAP attribute containing a URL or filename, you can select this option, then select your attribute.
- **Liberty User Profile:** If you have defined a Liberty User Profile attribute containing a URL or filename, you can select this option, then select your attribute.

Result on Condition Error: Specify what the condition returns when the comparison of the two values returns an error rather than the results of the comparison. Select either **False** or **True**. If you do not want the action applied when an error occurs, select **False**. If you want the action applied when an error occurs, select **True**.

URL File Extension Condition

The URL File Extension condition allows you to restrict access based on the file extension specified in the URL of the request. It compares the file extension in the URL of the current request to the extension specified in the **Value** field.

To set up matching for this condition, fill in the following fields:

Comparison: Select one of the following types:

- **Comparison: URL File:** Specifies that you want the values compared as file extensions and how you want the file extensions compared. Select one of the following:
  - **Equals:** Indicates that the file extensions must contain the same letters, in the same order as specified in the value.
  - **Starts with:** Indicates that the file extensions must begin with the letters specified in the value.
  - **Ends with:** Indicates that the file extensions must end with the letters specified in the value.
  - **Contains Substring:** Indicates that the file extensions must contain the letters, in the same sequence, as specified in the **Value** field.
- **Comparison: Regular Expression: Matches:** Specifies that you want the values compared as regular expressions.

Mode: Select the mode appropriate for the comparison type:

- **Comparison: URL File Extension:** Specify whether case is important by selecting **Case Sensitive** or **Case Insensitive**.
- **Comparison: Regular Expression: Matches:** Select one or more of the following:

  Canonical Equivalence
  Case Insensitive
  Comments

Dot All
Multi-Line
Unicode
Unix Lines

For regular expression syntax information, see the Javadoc for java.util.regex.Pattern.

Value: Specify the value type and value for the comparison. Select one of the following:

- **Data Entry Field:** To specify a static value to compare to the file extension in the current request, select this value type and specify the file extension. You can specify the extension or the period and the extension. For example:

  ```
  .ext
  ext
  ```

  This condition does not support wildcards. If you selected **URL File Extension** for the comparison type, you can add multiple values:

  - Use the **Edit** button to access a text box where you can enter multiple values, each on a separate line. For more information, see "Edit Button" on page 690.
  - Use the **Add** button to add values one at a time.

  All entered URL file extensions are compared to the requested URL file extension until a match is found or the list is exhausted.
- **LDAP Attribute:** If you have defined an LDAP attribute containing a URL or file extension, you can select this option, then select your attribute.
- **Liberty User Profile:** If you have defined a Liberty User Profile attribute containing a URL or file extension, you can select this option, then select your attribute.

Result on Condition Error: Specify what the condition returns when the comparison of the two values returns an error rather than the results of the comparison. Select either **False** or **True**. If you do not want the action applied when an error occurs, select **False**. If you want the action applied when an error occurs, select **True**.

Virtual Attribute Condition

The Virtual Attribute condition allows you to control access based on a value in an Virtual attribute. You can have the user's attribute value retrieved from an external source and compared to a value of the following type:

- Roles from an identity provider
- Authenticating IDP or user store
- Authentication contract, method, or type
- Credential profile
- LDAP attribute, OU, or group
- Liberty User Profile attribute
- Static value in a data entry field
- Virtual Attribute

To set up the matching for this condition, fill in the following fields:

Virtual Attribute: Specify the virtual attribute you want to use in the comparison. Select a virtual attribute from the list.'

Comparison: Specify how you want the values compared. All data types are available. Select one that matches the value type of your virtual attribute.

Mode: Select the mode, if available, that matches the comparison type. For example, if you select to compare the values as strings, you can select either a **Case Sensitive** mode or a **Case Insensitive** mode.

Value: Specify the second value for the comparison. All data types are available. For example, you can select to compare the value of one virtual attribute to the value of another virtual attribute. Only you can determine if such a comparison is meaningful.

Result on Condition Error: Specify what the condition returns when the comparison of the two values returns an error rather than the results of the comparison. Select either **False** or **True**. If you do not want the action applied when an error occurs, select **False**. If you want the action applied when an error occurs, select **True**.

X-Forwarded-For IP Condition

For added security, you can add the IP address of the reverse proxy as a condition to check before granting access. One way to implement this is to create a rule that requires the X-Forwarded-For IP address in the HTTP header to match the configured IP address of the reverse proxy that is using the policy. The X-Forwarded-For IP condition matches the first IP address in the X-Forwarded-For header with the IP address specified in the **Value** field.

To set up matching for this condition, fill in the following fields:

Comparison: Specify how the X-Forwarded-For IP address is compared to the data in the **Value** field. Select one of the following types:

- **Comparison: IP:** Specifies that you want the values compared as IP addresses. Select one of the following:
  - **Equals:** Allows you to specify an IP address that the X-Forwarded-For IP address must match. You can specify more than one.
  - **In Range:** Allows you to specify a range of IP addresses that the X-Forwarded-For IP address must fall within. You can specify more than one range.
  - **In Subnet:** Allows you to specify the subnet that the X-Forwarded-For IP address must belong to. You can specify more than one subnet.
- **Comparison: Regular Expression: Matches:** Specifies that you want the values compared as regular expressions. If you select this option, you must also specify a **Mode**. Select one or more of the following:

  Canonical Equivalence
  Case Insensitive
  Comments
  Dot All
  Multi-Line
  Unicode
  Unix Lines

  For regular expression syntax information, see the Javadoc for java.util.regex.Pattern.

Value: Specify the value type and value for the comparison. Select one of the following:

- **Client IP:** If you want the first IP address in the X-Forwarded-For header compared to the IP address of the client making the request, select this option.

NOTE: Client IP will not support IPv6 addresses.

- **LDAP Attribute:** If you have defined an LDAP attribute for an IP address, you can select this option, then select your attribute.
- **Liberty User Profile:** If you have defined a Liberty User Profile attribute for an IP address, you can select this option, then select your attribute.
- **X-Forwarded-For-IP:** Allows you to control access based on the value in the X-Forwarded-For IP header of the HTTP request. This supports IPv6 address when you use the X-Forwarded-For IP condition.
- **Data Entry Field:** To specify a static value, select **Data Entry Field** and provide a value appropriate for your comparison type. For example:

Comparison Type	IPv4 Value	IPv6 Value
Equals	10.10.10.10	2001:1000:1000:1000:1000:1000:1000:1a8a
	10.10.10.11	2001::10a0
In Range	10.10.10.10 - 10.10.10.100	2134::10 -2134::100
	10.10.20.10 - 10.10.20.100	2134:1000:1000:1000:1000:1000:2000:1000 - 2134:1000:1000:1000:1000:1000:2000:4000
In Subnet	10.10.10.12 / 22	2001:1000::0002:1000:1a8a/40
	10.10.20.30 / 22	2001:1000:1000:2000:3000:4000:5000:1a8a/50

You can now enter an IPv6 IP address. If you enter a zone ID and scope ID in an IP address with % sign, you will get an error. For more information see "Setting up L4 Switch for IPv6 Support" on page 779.

If you selected **IP** for the comparison type, you can add multiple values:

- Use the **Edit** button to access a text box where you can enter multiple values, each on a separate line.
- Use the **Add** button to add values one at a time.

All listed values are compared to the IP address in the X-Forwarded-For IP header until a match is found or the list is exhausted.

Result on Condition Error: Specify what the condition returns when the comparison of the two values returns an error rather than the results of the comparison. Select either **False** or **True**. If you do not want the action applied when an error occurs, select **False**. If you want the action applied when an error occurs, select **True**.

Condition Extension

If you have loaded and configured an authorization condition extension, this option specifies a condition that is evaluated by an outside source. This outside source returns either true or false. See the documentation that came with the extension for information about what is evaluated.

Data Extension

If you have loaded and configured an authorization data extension, this option specifies the value that the extension retrieves. You can then select to compare this value with an LDAP attribute, a Liberty User Profile attribute, a Data Entry Field, or another Data Extension. For more information, see the documentation that came with the extension.

Using the URL Dredge Option

In the **URL to Dredge** text box, enter the URL of a page on a Web server, then click **Display URL List**. A list of links and images appears.

For example, if you enter `www.netiq.com/documentation/index.html` for the **URL to Dredge**, links such as the following appear in the **Links** section of the **URL Results** list:

```
www.netiq.com/company/careers/index.html
www.netiq.com/company/strategy.html
www.netiq.com/documentation/netiqaccessmanager32/index.html
www.netiq.com/documentation/netiqaccessmanager31/index.html
```

Depending upon how you have configured your Web site, you need to enter either a target page or just the URL of the site to generate a list of links.

To add all links as values to the URL condition, click **Links**. To add links selectively as a value, select the check box next to each name. To dredge a link in the list, click the link.

If the URL contains images, a list of images appears in the **Images** section. To add an image as a value, select the check box next to the image name.

To save your changes, click **OK**.

IMPORTANT: If you attempt to dredge an HTTPS site that is using a self-signed certificate, you need to import the trusted root of the site into the Trusted Roots store of the Access Gateway before performing the dredge.

Edit Button

Some of the conditions such as Client IP and URL display an **Edit** button when you select **Equals** as the condition and **Data Entry Field** as the value. The **Edit** button displays a text box where you can specify multiple values.

In the text box, enter each value on a separate line.

To save your modifications, click **OK**.

To discard your modifications, click **Cancel**.

Importing and Exporting Authorization Policies

You can import and export Authorization policies in order to run them in other Access Manager Appliance configurations and to analyze the authorization logic. The policy is exported as a text file with XML tags. We do not recommend editing the exported file with a text editor. Any changes you want to make to a policy must be done through the Administration Console.

To export an Authorization policy:

1. In the Administration Console, click **Policies > Policies**.
2. Select an Authorization policy, then click **Export**.
3. (Optional) Modify the name suggested for the file.
4. Click **OK**.
5. Using the features of your browser, specify where you want the file to be copied.
6. Click **OK**.

To import a policy:

1. Ensure that any referenced Role policies have been imported.

   See “Importing and Exporting Role Policies” on page 642.
2. If the policy uses LDAP or Liberty Profile attributes, ensure that the Identity Server has been configured for these same attributes.
3. In the Administration Console, click **Policies** > **Policies**.
4. Click **Import**, then browse to and select the file.
5. Click **OK**.
6. When the policy appears in the list, click **Apply Changes**.

Identity Injection Policies

Identity injection allows you to add information to the URL or to the HTML page before it is posted to a Web server. The Web server uses this information to determine whether the user can access to the resource, so it is the Web server that determines the information that you need to inject to allow access to the resource.

Identity injection is one of the features of Access Manager Appliance that enable you to provide single sign-on for your users. When the policy is configured, the user is unaware that additional information is required to access a Web server.

IMPORTANT: Identity Injection policies allow you to inject the user’s password into the HTTP header. If you set up such a policy, you must also configure the Access Gateway to use SSL between itself and the back-end Web server. This is the only way to ensure that the password is encrypted on the wire.

This section describes the elements available for an Identity Injection policy, but your Web servers determine which elements you use.

- “Designing an Identity Injection Policy” on page 692
- “Configuring an Identity Injection Policy” on page 693
- “Configuring an Authentication Header Policy” on page 694
- “Configuring a Custom Header Policy” on page 698

- “Configuring a Custom Header with Tags” on page 701
- “Specifying a Query String for Injection” on page 703
- “Injecting into the Cookie Header” on page 706
- “Configuring an Inject Kerberos Ticket Policy” on page 706
- “Importing and Exporting Identity Injection Policies” on page 709
- “Sample Identity Injection Policy” on page 709

Designing an Identity Injection Policy

Before setting up an Identity Injection policy, you need to know the following about your Web application:

- Does it require an authentication header? Does this header need just the username or does it also need the password?
- Does it use a custom header with custom names (x-names)? If so, you need to know their names and their expected values.
- Does the custom header require any custom names (x-names) with tags? If so, gather this information.
- Does the application expect specific values in the query string of the URL? If so, gather this information.
- Does it require the authentication information from the Kerberos tickets? If so, gather this information.

After gathering the information, you need to determine whether you need to create one policy with one rule, one policy with multiple rules, or multiple policies. If you have multiple applications that require the same type of authentication header, you might want to create an authentication header policy and separate policies for the application-specific information. You can then enable both the authentication header policy and the application-specific policy for the resource that is protecting the application. You must design your policies so that the application receives just what it needs. It must not inject custom names and values it does not use.

Everything defined in a policy is injected into the header, even if the values are empty because Access Manager Appliance could not obtain the value for the item. For some applications, this is still useful information and the application uses it to make access decisions.

Whether you create a policy with one rule or multiple rules is a personal design decision. If you put all the actions in one rule, you have only one description field to describe the function of the policy. If you put each action type in a separate rule, you have multiple description fields to describe the function of the policy. Select the method that is easiest for you.

Rules are evaluated by priority. The first rule that is evaluated with an authentication header is processed, and the authentication header is rejected if it is found in any of the other rules. Your policy can inject only one authentication header, one cookie header, and one query string, but it can inject multiple custom headers and custom headers with tags.

Using the Refresh Data Option

Identity Injection policies are processed when a user requests access to a resource. The results and the values of the data items are cached for the user session. This means that when the user requests a second time to access the resource, the policy is evaluated, but the data values from the first

evaluation are used. When a data item is cached for the user session, the user must log out and log back in to trigger new data values. (For information about how long the data items are cached, see "The Policy Is Using Old User Data" on page 1007.)

The LDAP Attribute and the Shared Secret actions can be configured to refresh their values. This means the attribute or secret value is read not just on the first request that triggers the policy evaluation, but when the specified refresh interval expires. You can select to cache the value for the session, for the request, or for a time interval varying from 5 seconds to 60 minutes.

You can use this feature for situations when you do not want to force the user to log in again to gain rights to resources or to revoke rights to resources. For example, suppose that you have an Identity Injection policy that grants access based on an LDAP attribute in a custom header having a "yes" value. Users with a "no" value in custom header are denied access.

If you don't enable the Refresh Data option on this attribute in the policy, the policy is evaluated when the user first tries to access the resource. The value for the attribute is cached for the user session, and until the user logs out, that is the value that is used.

However, if you enable the Refresh Data option on this attribute in the policy, the policy is evaluated when the user first tries to access the resource. When the user sends a second request to access the resource and the specified interval has expired, the Refresh Data option causes the value of the attribute to be read again from the LDAP server. This new value is injected into the custom header, and any other policy that is triggered by the request and uses the new value for its policy.

- If the value from the first request to the second request changes from no to yes, the user gets access to the resource.
- If the value from the first request to the second request changes from yes to no, the user is denied access to the resource.

For example:

- If the attribute controls access to employee resources and an employee leaves, a quick change of this attribute value cuts the employee off from the resources that must be available to employees only.
- If the attribute controls access to a software download site and a user has just purchased a product, a quick change to this attribute value can grant access to the download site.

IMPORTANT: This feature needs to be used with caution. Because querying the LDAP server slows down the processing of a policy, LDAP attribute and secret store values are normally cached for the user session. Enable this option only on those attributes and secrets that are critical to the security of your system or to the design of your work flow.

Configuring an Identity Injection Policy

1. In the Administration Console, click **Policies > Policies**.
2. Select the policy container, then click **New.**
3. Specify a name for the policy, select **Access Gateway: Identity Injection** for the type of policy, then click **OK**.
4. Fill in the following fields:

   Description: (Optional) Describe the purpose of this policy. Because Identity Injection policies are customized to match the content of a specific Web server, you might want to include the name of the Web server as part of the description.

Priority: Specify the order in which a rule is applied in the policy, when the policy has multiple rules. The highest priority is 1 and the lowest priority is 10.

5 In the **Actions** section, click **New**, then select one of the following.

- **Inject into Authentication Header:** Inserts the username and password into the header. For information about how to configure this type of policy, see "Configuring an Authentication Header Policy" on page 694.
- **Inject into Custom Header:** Inserts custom names with values into the custom header. For information about how to configure this type of policy, see "Configuring a Custom Header Policy" on page 698.
- **Inject into Custom Header with Tags:** Inserts custom tags with name/value content into the custom header. For information about how to configure this type of policy, see "Configuring a Custom Header with Tags" on page 701.
- **Inject into Query String:** Inserts a query string into the URL for the page. For information about how to configure this type of policy, see "Specifying a Query String for Injection" on page 703.
- **Inject into Cookie Header:** Inserts the session cookie into the cookie header. For information about how to configure this type of policy, see "Injecting into the Cookie Header" on page 706.
- **Inject Kerberos Ticket:** Inserts authentication values from the Kerberos ticket into the custom header. For information about how to configure this type of policy, see "Configuring an Inject Kerberos Ticket Policy" on page 706.

6 (Optional) Repeat Step 5.

Repeat this process to add multiple actions to the same rule. If a particular action is allowed only once per rule, then the action does not appear in the **New** menu if that action has already been defined in the rule. If an action is allowed multiple times per rule, you can select it from the **New** menu or use the **Copy Action** icon and modify the new entry.

7 To save the policy, click **OK** twice, then click **Apply Changes**.

8 For information about how to assign the policy to a protected resource, see "Assigning an Identity Injection Policy to a Protected Resource" on page 107.

Configuring an Authentication Header Policy

To inject values into the authentication header, you need to know what the Web server requires. For basic authentication, you need to inject the username and password. For a sample policy for a Web server that requires the LDAP username and password to be injected into the header, see "Setting Up Policies" on page 88.

To create and configure an authentication header policy:

1 In the Administration Console, click **Policies > Policies**.

2 Select the policy container, then click **New**.

3 Specify a name for the policy, select **Access Gateway: Identity Injection** for the type, then click **OK**.

4 (Optional) Specify a description for the injection policy. This is useful if you plan to create multiple policies to be used by multiple resources.

5 In the **Actions** section, click **New**, then select **Inject into Authentication Header**.

6 Fill in the **User Name** field.

Select **Credential Profile** to insert the name the user entered when the user authenticated. This is the most common value type to use for the username.

The default contracts assign the cn attribute to the Credential Profile. If you have created a custom contract that uses credentials other than the ones listed below, do not use the Credential Profile as a condition.

If your user store is an Active Directory server, the SAMAccountName attribute is used for the username and stored in the cn field of the LDAP Credential Profile.

Depending upon what the user must supply for authentication, select one of the following:

- **LDAP Credentials:** If you prompt the user for a username, select this option, then select either **LDAP User Name** (the cn attribute of the user) or **LDAP User DN** (the fully distinguished name of the user). Your Web server requirements determine which one you use.
- **X509 Credentials:** If you prompt the user for a certificate, select this option, then select one of the following options depending upon your Web server requirements.
  - **X509 Public Certificate Subject:** Injects just the subject field from the certificate, which can match the DN of the user, depending upon who issued the certificate.
  - **X509 Public Certificate Issuer:** Injects just the issuer field from the certificate, which is the name of the certificate authority (CA) that issued the certificate.
  - **X509 Public Certificate:** Injects the entire certificate.
  - **X509 Serial Number:** Injects the certificate serial number.
- **SAML Credential:** Although this option is available for the username, most applications that use SAML assertions use them for the user's password. For the username, you must probably select an option that allows you to supply the user's name, such as **LDAP Credentials** or **LDAP Attribute**.

Your Web server requirements determine the data type you select for the username. LDAP, X509, and SAML credentials are available from the Credential Profile. If you have created a custom contract that uses a credential other than the ones listed in the Credential Profile, you can select one of the following values to insert into the header as the username:

- **Authentication Contract:** Injects the URI of the authentication contract the user used for authentication.
- **Client IP:** Injects the IP address associated with the user.
- **LDAP Attribute:** Injects the value of the selected attribute. For Active Directory servers, specify the SAMAccountName attribute for the username. If the attribute you require does not appear in the list, click **New LDAP Attribute** to add the attribute.

  The **Refresh Data Every** option allows you to determine when to send a query to the LDAP server to verify the current value of the attribute. Because querying the LDAP server slows down the processing of a policy, LDAP attribute values are normally cached for the user session.

  Change the value of this option from session to a more frequent interval only on those attributes that are critical to the security of your system or to the design of your work flow. You can select to cache the value for the session, for the request, or for a time interval varying from 5 seconds to 60 minutes.

  For more information, see "Using the Refresh Data Option" on page 692.
- **Liberty User Profile:** Injects the value of the selected attribute. If no profile attributes are available, you have not enabled their use in the Identity Server configuration. See "Managing Web Services and Profiles" on page 462.
- **Proxy Session Cookie:** Injects the session cookie associated with the user.
- **Roles:** Injects the roles that have been assigned to the user.
- **Shared Secret:** Injects the username that has been stored in the selected shared secret store.

You can create your own username attribute. Click **New Shared Secret**, specify a display name for the store, and Access Manager Appliance creates the store. Select the store, click **New Shared Secret Entry**, specify a name for the attribute, then click **OK**. The store can contain one name/value pair or a collection of name/value pairs. For more information, see "Creating and Managing Shared Secrets" on page 732.

The **Refresh Data Every** option allows you to determine when to send a query to verify the current value of the secret. Because querying slows down the processing of a policy, secret values are normally cached for the user session.

Change the value of this option from session to a more frequent interval only on those secrets that are critical to the security of your system or to the design of your work flow. You can select to cache the value for the session, for the request, or for a time interval varying from 5 seconds to 60 minutes. For more information, see "Using the Refresh Data Option" on page 692.

- **Virtual Attribute:** Injects the value of the selected virtual attribute.

  The **Refresh Data Every** option allows you to determine when to send a query to verify the current value of the virtual attribute. Because querying slows down the processing of a policy, the virtual attribute values are normally cached for the user session.

  Change the value of this option from session to a more frequent interval only on those attributes that are critical to the security of your system or to the design of your work flow. You can select to cache the value for the session, for the request, or for a time interval varying from 5 seconds to 60 minutes. For more information, see "Using the Refresh Data Option" on page 692.

- **X-Forwarded-For IP:** Injects the X-Forwarded-For IP address of the client.
- **String Constant:** Injects a static value that you specify in the text box. This value is used by all users who access the resources assigned to this policy.
- **Data Extension:** (Conditional) If you have installed a data extension for Identity Injection policies, this option injects the value that the extension retrieves. For more information about creating a data extension, see Access Manager SDK Sample Code.

The value type you use depends upon how you have set up the application.

NOTE: To improve the policy's performance, configure the LDAP Attributes, Credential Profile, Liberty User Profile, and Shared Secret attributes to be sent with authentication. For more information, see "Configuring the Attributes Sent with Authentication" on page 156.

---

7 Fill in the **Password** field.

Select **Credential Profile** to insert the password the user entered when the user authenticated. This is the most common value type to use for the password. If you have created a custom contract that uses credentials other than the ones listed below for the password, do not use the Credential Profile for the password.

- **LDAP Credentials:** If you prompt the user for a password, select this option, then select **LDAP Password**. If the user's password is the same as the name of the user, you can select either **LDAP User Name** (the cn attribute of the user) or **LDAP User DN** (the fully distinguished name of the user).
- **X509 Credentials:** If you use a certificate for the password, select this option, then select one of the following:
  - **X509 Public Certificate Subject:** Injects just the subject from the certificate, which can match the DN of the user, depending upon who issued the certificate.
  - **X509 Public Certificate Issuer:** Injects just the issuer from the certificate, which is the name of the certificate authority (CA) that issued the certificate.
  - **X509 Public Certificate:** Injects the entire certificate.

- **X509 Serial Number:** Injects the certificate serial number.

- **SAML Credential:** Injects the SAML assertion in the authentication header as the user's password.

Your Web server requirements determine the data type you select for the password. LDAP, X509, and SAML credentials are available from the Credential Profile. You can also select one of the following values to insert into the header as the password:

- **Authentication Contract:** Injects the URI of a local authentication contract that the user used for authentication.
- **Client IP:** Injects the IP address associated with the user.
- **LDAP Attribute:** Injects the value of the selected attribute. For Active Directory servers, specify the SAMAccountName attribute for the username. If the attribute you require does not appear in the list, click **New LDAP Attribute** to add the attribute.

  The **Refresh Data Every** option allows you to determine when to send a query to the LDAP server to verify the current value of the attribute. Because querying the LDAP server slows down the processing of a policy, LDAP attribute values are normally cached for the user session.

  Change the value of this option from session to a more frequent interval only on those attributes that are critical to the security of your system or to the design of your work flow. You can select to cache the value for the session, for the request, or for a time interval varying from 5 seconds to 60 minutes. For more information, see "Using the Refresh Data Option" on page 692.
- **Liberty User Profile:** Injects the value of the selected attribute.
- **Proxy Session Cookie:** Injects the session cookie associated with the user.
- **Roles:** Injects the roles that have been assigned to the user.
- **Shared Secret:** Injects the password that has been stored in the selected shared secret store.

  You can create your own password attribute. Click **New Shared Secret**, specify a display name for the store, and the Access Manager Appliance creates the store. Select the store, click **New Shared Secret Entry**, specify a name for the attribute, then click **OK**. The store can contain one name/value pair or a collection of name/value pairs. For more information, see "Creating and Managing Shared Secrets" on page 732.

  The **Refresh Data Every** option allows you to determine when to send a query to verify the current value of the secret. Because querying slows down the processing of a policy, secret values are normally cached for the user session.

  Change the value of this option from session to a more frequent interval only on those secrets that are critical to the security of your system or to the design of your work flow. You can select to cache the value for the session, for the request, or for a time interval varying from 5 seconds to 60 minutes. For more information, see "Using the Refresh Data Option" on page 692.
- **X-Forwarded-For IP:** Injects the X-Forwarded-For IP address of the client.
- **Virtual Attribute:** Injects the value of the selected virtual attribute.

  The **Refresh Data Every** option allows you to determine when to send a query to verify the current value of the virtual attribute. Because querying slows down the processing of a policy, the virtual attribute values are normally cached for the user session.

Change the value of this option from session to a more frequent interval only on those attributes that are critical to the security of your system or to the design of your work flow. You can select to cache the value for the session, for the request, or for a time interval varying from 5 seconds to 60 minutes. For more information, see “Using the Refresh Data Option” on page 692.

- **String Constant:** Injects a static value that you specify in the text box. This value is used by all users who access the resources assigned to this policy.
- **Data Extension:** (Conditional) If you have installed a data extension for Identity Injection policies, this option injects the value that the extension retrieves. For more information about creating a data extension, see NetIQ Access Manager Developer Tools and Examples.

The value type you use depends upon how you have set up the application.

8 Specify the format for the value:

Multi-Value Separator: Select a value separator, if the value type you have select is multi-valued. For example, **Roles** can contain multiple values.

DN Format: If the value is a DN, select the format for the DN:

- **LDAP:** Specifies LDAP typed comma notation:

  ```
  cn=jsmith,ou=Sales,o=novell
  ```

- **NDAP Partial Dot Notation:** Specifies eDirectory™ typeless dot notation.

  ```
  jsmith.sales.novell
  ```

- **NDAP Leading Partial Dot Notation:** Specifies eDirectory typeless leading dot notation.

  ```
  .jsmith.sales.novell
  ```

- **NDAP Fully Qualified Partial Dot Notation:** Indicates eDirectory typed dot notation.

  ```
  cn=jsmith.ou=Sales.o=novell
  ```

- **NDAP Fully Qualified Leading Dot Notation:** Indicates eDirectory typed leading dot notation.

  ```
  .cn=jsmith.ou=Sales.o=novell
  ```

9 Click **OK**.

10 (Optional) To add a second rule, click **New** in the Rule List.

You can inject only one authentication header into an Identity Injection rule. However, your policy can have multiple rules. If you inject two authentication headers, each in a separate rule, the authentication header in the rule with the highest priority is applied, and the authentication header action in the second rule is ignored.

11 To save the policy, click **OK**, then click **Apply Changes**.

Configuring a Custom Header Policy

To inject values into a custom header, you need to know the name of the tag and its expected value type. The names are specific to the application. The names might be case sensitive. They might require an X- prefix. Because the requirements vary, you need to enter them in the format as specified by the application. For example, an application might require the following to be in the custom header:

Name/Value Pair	Description
`X-First_Name=givenName`	A first name tag with an LDAP attribute value
`X-Last_Name=sn`	A last name tag with an LDAP attribute value
`X-Role=sales_role`	A role tag with the role name as the value.

If you create a custom header policy with these name/value pairs, the policy injects these names with their values into a custom header, before sending the request to the Web server.

To create such a policy:

1. In the Administration Console, click **Policies > Policies**.
2. Select the policy container, then click **New.**
3. Specify a name for the policy, select **Access Gateway: Identity Injection** for the type, then click **OK**.
4. (Optional) Specify a description for the injection policy. This is useful if you plan to create multiple custom header policies to be used for multiple resources.
5. In the **Actions** section, click **New**, then select **Inject into Custom Header**.
6. Fill in the following fields:

   Custom Header Name: Specify the name to be inserted into the custom header. These are the names required by your application. If your application requires the X- prefix, ensure that you include the prefix in this field.

   Value: Select the value required by the name. Select one of the following:

   - **Authentication Contract:** Injects the URI of a local authentication contract that the user used for authentication.
   - **Client IP:** Injects the IP address associated with the user.
   - **Credential Profile:** Injects the credentials that the user specified at login. You can select **LDAP Credentials**, **X509 Credentials**, or **SAML Credentials**. For more information, see “Configuring an Authentication Header Policy” on page 694.
   - **LDAP Attribute:** Injects the value of the selected attribute. For Active Directory servers, specify the SAMAccountName attribute for the username. If the attribute you require does not appear in the list, click **New LDAP Attribute** to add the attribute.

     The **Refresh Data Every** option allows you to determine when to send a query to the LDAP server to verify the current value of the attribute. Because querying the LDAP server slows down the processing of a policy, LDAP attribute values are normally cached for the user session.

     Change the value of this option from session to a more frequent interval only on those attributes that are critical to the security of your system or to the design of your work flow. You can select to cache the value for the session, for the request, or for a time interval varying from 5 seconds to 60 minutes.

     For more information, see “Using the Refresh Data Option” on page 692.
   - **Liberty User Profile:** Injects the value of the selected attribute. If no profile attributes are available, you have not enabled their use in the Identity Server configuration. See “Managing Web Services and Profiles” on page 462.
   - **Proxy Session Cookie:** Injects the session cookie associated with the user.
   - **Roles:** Injects the roles that have been assigned to the user.

- **Shared Secret:** Injects a value that has been stored in the selected shared secret store. Select the shared secret store and the name of the value you want injected.

  You can create your own value. Click **New Shared Secret**, specify a display name for the store, and the Access Manager Appliance creates the store. Select the store, click **New Shared Secret Entry**, specify a name for the attribute, then click **OK**. The name you select for the attribute must match the Custom Header name. The store can contain one name/value pair or a collection of name/value pairs. For more information, see "Creating and Managing Shared Secrets" on page 732.

  The **Refresh Data Every** option allows you to determine when to send a query to verify the current value of the secret. Because querying slows down the processing of a policy, secret values are normally cached for the user session.

  Change the value of this option from session to a more frequent interval only on those secrets that are critical to the security of your system or to the design of your work flow. You can select to cache the value for the session, for the request, or for a time interval varying from 5 seconds to 60 minutes. For more information, see "Using the Refresh Data Option" on page 692.

- **Virtual Attribute:** Injects the value of the selected virtual attribute.

  The **Refresh Data Every** option allows you to determine when to send a query to verify the current value of the virtual attribute. Because querying slows down the processing of a policy, the virtual attribute values are normally cached for the user session.

  Change the value of this option from session to a more frequent interval only on those attributes that are critical to the security of your system or to the design of your work flow. You can select to cache the value for the session, for the request, or for a time interval varying from 5 seconds to 60 minutes. For more information, see "Using the Refresh Data Option" on page 692.

- **X-Forwarded-For IP:** Injects the X-Forwarded-For IP address of the client.
- **String Constant:** Injects a static value that you specify in the text box. This value is used by all users who access the resources assigned to this policy.
- **Data Extension:** (Conditional) If you have installed a data extension for Identity Injection policies, this option injects the value that the extension retrieves. For more information about creating a data extension, see NetIQ Access Manager Developer Tools and Examples.

NOTE: To improve the policy's performance, configure the LDAP Attributes, Credential Profile, Liberty User Profile, and Shared Secret attributes to be sent with authentication. For more information, see "Configuring the Attributes Sent with Authentication" on page 156.

---

7 Specify the format for the value:

Multi-Value Separator: Select a value separator, if the value type you have select is multi-valued. For example, **Roles** can contain multiple values.

DN Format: If the value is a DN, select the format for the DN:

- **LDAP:** Specifies LDAP typed comma notation.

  ```
  cn=jsmith,ou=Sales,o=novell
  ```

- **NDAP Partial Dot Notation:** Specifies eDirectory typeless dot notation.

  ```
  jsmith.sales.novell
  ```

- **NDAP Leading Partial Dot Notation:** Specifies eDirectory typeless leading dot notation.

  ```
  .jsmith.sales.novell
  ```

- **NDAP Fully Qualified Partial Dot Notation:** Indicates eDirectory typed dot notation.

  ```
  cn=jsmith.ou=Sales.o=novell
  ```

- **NDAP Fully Qualified Leading Dot Notation:** Indicates eDirectory typed leading dot notation.

  ```
  .cn=jsmith.ou=Sales.o=novell
  ```

8. (Optional) To add additional custom header actions, click **New**, then select **Inject into Custom Header** or use the **Copy Action** icon  and modify the new entry.
9. To save the policy, click **OK** twice, then click **Apply Changes**.

Configuring a Custom Header with Tags

Some Web applications require more than a name and a value to be injected into the custom header. Sometimes they require a custom name, a tag, and a value. Sometimes the application requires a custom name with multiple tags and values. The **Inject into Custom Header with Tags** option provides you with the flexibility to add such values to the custom header. For example, your application could be expecting the following custom header with tag:

```
X-Custom_Role Role=Manager
```

You can inject this information by setting the **Custom Header Name** to X-Custom, the **Tag Name** to Role, and the **Tag Value** to Manager. The value can be set as a static variable or you can retrieve it from various sources such as a Liberty User Profile attribute or the roles assigned to the current user.

1. In the Administration Console, click **Policies > Policies**.
2. Select the policy container, then click **New**.
3. Specify a name for the policy, select **Access Gateway: Identity Injection** for the type, then click **OK**.
4. (Optional) Specify a description for the injection policy. This is useful if you plan to create multiple custom header policies to be used for multiple resources.
5. In the **Actions** section, click **New**, then select **Inject into Custom Header with Tags**.
6. Fill in the following fields:

   Custom Header Name: Specify the name that the application expects. If your application requires the X- prefix, ensure that you include the prefix in this field.

   Tag Name: Specify the tag name that the application expects.

   Tag Value: Specify the value. Select from the following data types:

   - **Authentication Contract:** Injects the URI of a local authentication contract that the user used for authentication.
   - **Client IP:** Injects the IP address associated with the user.
   - **Credential Profile:** Injects the credentials that the user specified at login. You can select **LDAP Credentials, > X509 Credentials**, or **SAML Credential**. For more information, see “Configuring an Authentication Header Policy” on page 694.
   - **LDAP Attribute:** Injects the value of the selected attribute. For Active Directory servers, specify the SAMAccountName attribute for the username. If the attribute you require does not appear in the list, click **New LDAP Attribute** to add the attribute.

The **Refresh Data Every** option allows you to determine when to send a query to the LDAP server to verify the current value of the attribute. Because querying the LDAP server slows down the processing of a policy, LDAP attribute values are normally cached for the user session.

Change the value of this option from session to a more frequent interval only on those attributes that are critical to the security of your system or to the design of your work flow. You can select to cache the value for the session, for the request, or for a time interval varying from 5 seconds to 60 minutes. For more information, see "Using the Refresh Data Option" on page 692.

- **Liberty User Profile:** Injects the value of the selected attribute. If no profile attributes are available, you have not enabled their use in the Identity Server configuration. See "Managing Web Services and Profiles" on page 462.
- **Proxy Session Cookie:** Injects the session cookie associated with the user.
- **Roles:** Injects the roles that have been assigned to the user.
- **Shared Secret:** Injects a value that has been stored in the selected shared secret store. The name specified as the Tag Name must match the name of a name/value pair stored in the shared secret.

  You can create your own value. Click **New Shared Secret**, specify a display name for the store, and the Access Manager Appliance creates the store. Select the store, click **New Shared Secret Entry**, specify a name for the attribute, then click **OK**. The name must match the expected Tag Name. The store can contain one name/value pair or a collection of name/value pairs. For more information, see "Creating and Managing Shared Secrets" on page 732.

  The **Refresh Data Every** option allows you to determine when to send a query to verify the current value of the secret. Because querying slows down the processing of a policy, secret values are normally cached for the user session.

  Change the value of this option from session to a more frequent interval only on those secrets that are critical to the security of your system or to the design of your work flow. You can select to cache the value for the session, for the request, or for a time interval varying from 5 seconds to 60 minutes. For more information, see "Using the Refresh Data Option" on page 692.
- **Virtual Attribute:** Injects the value of the selected virtual attribute.

  The **Refresh Data Every** option allows you to determine when to send a query to verify the current value of the virtual attribute. Because querying slows down the processing of a policy, the virtual attribute values are normally cached for the user session.

  Change the value of this option from session to a more frequent interval only on those attributes that are critical to the security of your system or to the design of your work flow. You can select to cache the value for the session, for the request, or for a time interval varying from 5 seconds to 60 minutes. For more information, see "Using the Refresh Data Option" on page 692.
- **X-Forwarded-For IP:** Injects the X-Forwarded-For IP address of the client.
- **String Constant:** Injects a static value that you specify in the text box. This value is used by all users who access the resources assigned to this policy.
- **Data Extension:** (Conditional) If you have installed a data extension for Identity Injection policies, this option injects the value that the extension retrieves. For more information about creating a data extension, see NetIQ Access Manager Developer Tools and Examples.

NOTE: To improve the policy's performance, configure the LDAP Attributes, Credential Profile, Liberty User Profile, and Shared Secret attributes to be sent with authentication. For more information, see "Configuring the Attributes Sent with Authentication" on page 156.

7 To add multiple tag and value pairs to the custom name, click **New** in the **Tags** section.

Use the up-arrow and down-arrow buttons to order the tags.

8 Specify the format for the value:

Multi-Value Separator: Select a value separator, if the value type you have select is multi-valued. For example, **Roles** can contain multiple values.

DN Format: If the value is a DN, select the format for the DN:

- **LDAP:** Specifies LDAP typed comma notation.

  ```
  cn=jsmith,ou=Sales,o=novell
  ```

- **NDAP Partial Dot Notation:** Specifies eDirectory typeless dot notation.

  ```
  jsmith.sales.novell
  ```

- **NDAP Leading Partial Dot Notation:** Specifies eDirectory typeless leading dot notation.

  ```
  .jsmith.sales.novell
  ```

- **NDAP Fully Qualified Partial Dot Notation:** Indicates eDirectory typed dot notation.

  ```
  cn=jsmith.ou=Sales.o=novell
  ```

- **NDAP Fully Qualified Leading Dot Notation:** Indicates eDirectory typed leading dot notation.

  ```
  .cn=jsmith.ou=Sales.o=novell
  ```

9 (Optional) To add additional custom header actions, click **New**, then select **Inject into Custom Header with Tags** or use the **Copy Action** icon  and modify the new entry.

10 To save the policy, click **OK** twice, then click **Apply Changes**.

Specifying a Query String for Injection

Some applications require custom information in a query string of the URL. The **Inject into Query String** option allows you to inject this information without prompting the user for it. To inject the information, you must specify a tag name and a tag value. The tag name is what your application requires. For example, suppose your application expects the following query string for user jsmith:

```
?name=jsmith
```

You can inject this information into the URL by specifying a name for the **Tag Name** and **Credential Profile** for the **Tag Value**. The **Credential Profile** value type inserts the name that the current user specified when authenticating to the Access Gateway.

1 In the Administration Console, click **Policies > Policies**.

2 Select the policy container, then click **New**.

3 Specify a name for the policy, select **Access Gateway: Identity Injection** for the type, then click **OK**.

4 (Optional) Specify a description for the injection policy.

5 In the **Actions** section, click **New**, then select **Inject into Query String**.

6 Fill in the following fields:

Tag Name: Specify the tag name that the application expects.

Tag Value: Specify the value. Select from the following data types:

- **Authentication Contract:** Injects the URI of a local authentication contract that the user used for authentication.
- **Client IP:** Injects the IP address associated with the user.
- **Credential Profile:** Injects the credentials that the user specified at login. You can select **LDAP Credentials, > X509 Credentials**, or **SAML Credential**. For more information, see "Configuring an Authentication Header Policy" on page 694.
- **LDAP Attribute:** Injects the value of the selected attribute. For Active Directory servers, specify the SAMAccountName attribute for the username. If the attribute you require does not appear in the list, click **New LDAP Attribute** to add the attribute.

  The **Refresh Data Every** option allows you to determine when to send a query to the LDAP server to verify the current value of the attribute. Because querying the LDAP server slows down the processing of a policy, LDAP attribute values are normally cached for the user session.

  Change the value of this option from session to a more frequent interval only on those attributes that are critical to the security of your system or to the design of your work flow. You can select to cache the value for the session, for the request, or for a time interval varying from 5 seconds to 60 minutes. For more information, see "Using the Refresh Data Option" on page 692.
- **Liberty User Profile:** Injects the value of the selected attribute. If no profile attributes are available, you have not enabled their use in the Identity Server configuration. See "Managing Web Services and Profiles" on page 462.
- **Proxy Session Cookie:** Injects the session cookie associated with the user.
- **Roles:** Injects the roles that have been assigned to the user.
- **Shared Secret:** Injects a value that has been stored in the selected shared secret store. The name specified as the Tag Name must match the name of a name/value pair stored in the shared secret.

  You can create your own value. Click **New Shared Secret**, specify a display name for the store, and the Access Manager Appliance creates the store. Select the store, click **New Shared Secret Entry**, specify a name for the attribute, then click **OK**. The name you specify must match the Tag Name. The store can contain one name/value pair or a collection of name/value pairs. For more information, see "Creating and Managing Shared Secrets" on page 732.

  The **Refresh Data Every** option allows you to determine when to send a query to verify the current value of the secret. Because querying slows down the processing of a policy, secret values are normally cached for the user session.

  Change the value of this option from session to a more frequent interval only on those secrets that are critical to the security of your system or to the design of your work flow. You can select to cache the value for the session, for the request, or for a time interval varying from 5 seconds to 60 minutes. For more information, see "Using the Refresh Data Option" on page 692.
- **Virtual Attribute:** Injects the value of the selected virtual attribute.

  The **Refresh Data Every** option allows you to determine when to send a query to verify the current value of the virtual attribute. Because querying slows down the processing of a policy, the virtual attribute values are normally cached for the user session.

Change the value of this option from session to a more frequent interval only on those attributes that are critical to the security of your system or to the design of your work flow. You can select to cache the value for the session, for the request, or for a time interval varying from 5 seconds to 60 minutes. For more information, see “Using the Refresh Data Option” on page 692.

- **X-Forwarded-For IP:** Injects the X-Forwarded-For IP address of the client.
- **String Constant:** Injects a static value that you specify in the text box. This value is used by all users who access the resources assigned to this policy.
- **Data Extension:** (Conditional) If you have installed a data extension for Identity Injection policies, this option injects the value that the extension retrieves. For more information about creating a data extension, see NetIQ Access Manager Developer Tools and Examples.

NOTE: To improve the policy's performance, configure the LDAP Attributes, Credential Profile, Liberty User Profile, and Shared Secret attributes to be sent with authentication. For more information, see “Configuring the Attributes Sent with Authentication” on page 156.

---

7 (Optional) To add multiple tag and value pairs, click **New** in the **Tags** section.

You can inject only one query string into a rule, but you can inject multiple tag-name and tag-value pairs in the single query string.

Use the up-arrow and down-arrow buttons to order the tags.

8 Specify the format for the values:

Multi-Value Separator: Select a value separator, if the value type you have select is multi-valued. For example, **Roles** can contain multiple values.

DN Format: If the value is a DN, select the format for the DN:

- **LDAP:** Specifies LDAP typed comma notation.

  ```
  cn=jsmith,ou=Sales,o=novell
  ```

- **NDAP Partial Dot Notation:** Specifies eDirectory typeless dot notation.

  ```
  jsmith.sales.novell
  ```

- **NDAP Leading Partial Dot Notation:** Specifies eDirectory typeless leading dot notation.

  ```
  .jsmith.sales.novell
  ```

- **NDAP Fully Qualified Partial Dot Notation:** Indicates eDirectory typed dot notation.

  ```
  cn=jsmith.ou=Sales.o=novell
  ```

- **NDAP Fully Qualified Leading Dot Notation:** Indicates eDirectory typed leading dot notation.

  ```
  .cn=jsmith.ou=Sales.o=novell
  ```

9 To save the policy, click **OK** twice, then click **Apply Changes**.

Injecting into the Cookie Header

Some applications require access to the Access Gateway session cookie and expect to find it in the cookie header. You can create an Identity Injection policy that adds this cookie to the cookie header.

1. In the Administration Console, click **Policies > Policies**.
2. Select the policy container, then click **New**.
3. Specify a name for the policy, select **Access Gateway: Identity Injection** for the type, then click **OK**.
4. (Optional) Specify a description for the injection policy.
5. In the **Actions** section, click **New**, then select **Inject into Cookie Header**.

   This action allows only one value unless you have installed a data extension. If you have installed a data extension, you can select either **Proxy Session Cookie** or the **Data Extension**.

   Proxy Session Cookie: Injects the session cookie for the user.

   Data Extension: Injects the value retrieved from the extension. For more information about creating a data extension, see NetIQ Access Manager Developer Tools and Examples.
6. To save the policy, click **OK** twice, then click **Apply Changes**.

Configuring an Inject Kerberos Ticket Policy

This policy allows the authentication information in the Kerberos tickets to be passed to the Access Gateway.

To create and configure an Inject Kerberos Ticket policy, perform the following steps:

1. In the Administration Console, click **Policies** > **Policies**.
2. Select the policy container, then click **New**.
3. Specify a name for the policy, select **Access Gateway: Identity Injection** for the type, then click **OK**.
4. (Optional) Specify a description for the injection policy. This is useful if you plan to create multiple policies to be used by multiple resources.
5. In the **Actions** section, click **New**, then select **Inject Kerberos Ticket**.
6. Select an appropriate option in **User Name**.

   Select **Credential Profile** to insert the name a user enters during the authentication process.

   This is the most common value. The default contracts assign the cn attribute to the Credential Profile.

   If your user store is an Active Directory server, the SAMAccountName attribute is used for the username and stored in the cn field of the LDAP Credential Profile.

   Depending upon what a user must supply for authentication, select one of the following:

   - **LDAP Credentials:** If you prompt the user for a username, select this option. Then select either **LDAP User Name** (the cn attribute of the user) or **LDAP User DN** (the fully distinguished name of the user). Your Web server requirements determine which one you use.
   - **X509 Credentials:** If you prompt the user for a certificate, select this option. Then select one of the following options depending upon your Web server requirements:
     - **X509 Public Certificate Subject:** Injects the subject field from the certificate, which can match the DN of the user, depending upon who issued the certificate.

- **X509 Public Certificate Issuer:** Injects the issuer field from the certificate, which is the name of the certificate authority (CA) that issued the certificate.
- **X509 Public Certificate:** Injects the entire certificate.
- **X509 Serial Number:** Injects the certificate serial number.

- **SAML Credential:** Although this option is available for the username, most applications that use SAML assertions, use them for the user's password. For the username, select an option that allows you to supply the user's name, such as **LDAP Credentials** or **LDAP Attribute**.

Your Web server requirements determine the data type you select for the username. LDAP, X509, and SAML credentials are available from the Credential Profile. If you have created a custom contract that uses a credential other than the ones listed in the Credential Profile, you can select one of the following values to insert into the Kerberos ticket as the username:

- **Authentication Contract:** Injects the URI of the authentication contract the user specified for authentication.
- **Client IP:** Injects the IP address associated with the user.
- **LDAP Attribute:** Injects the value of the selected attribute. For Active Directory servers, specify the SAMAccountName attribute for the username. If the attribute you require does not appear in the list, click **New LDAP Attribute** to add the attribute.

  The **Refresh Data Every** option allows you to determine when to send a query to the LDAP server to verify the current value of the attribute. Because querying the LDAP server slows down the processing of a policy, LDAP attribute values are cached for a user session.

  Change the value of this option from session to a more frequent interval only on those attributes that are critical to the security of your system or to the design of your work flow. You can select to cache the value for the session, for the request, or for a time interval varying from 5 seconds to 60 minutes.

  For more information, see "Using the Refresh Data Option" on page 692.
- **Liberty User Profile:** Injects the value of the selected attribute. If no profile attributes are available, you have not enabled their use in the Identity Server configuration. See "Managing Web Services and Profiles" on page 462.
- **Proxy Session Cookie:** Injects the session cookie associated with the user.
- **Roles:** Injects the roles that have been assigned to the user.
- **Shared Secret:** Injects the username that has been stored in the selected shared secret store.

  You can create your own username attribute.

  1. Click **New Shared Secret**, specify a display name for the store, and Access Manager Appliance creates a store.
  2. Select the store, click **New Shared Secret Entry**, specify a name for the attribute, then click **OK**.

     The store can contain one name/value pair or a collection of name/value pairs. For more information, see "Creating and Managing Shared Secrets" on page 732.

  The **Refresh Data Every** option allows you to determine when to send a query to verify the current value of the secret. Because querying slows down the processing of a policy, secret values are cached for a user session.

  Change the value of this option from session to a more frequent interval only on those secrets that are critical to the security of your system or to the design of your work flow. You can select to cache the value for the session, for the request, or for a time interval varying from 5 seconds to 60 minutes. For more information, see "Using the Refresh Data Option" on page 692.

- **Virtual Attribute:** Injects the value of the selected virtual attribute.

  The **Refresh Data Every** option allows you to determine when to send a query to verify the current value of the virtual attribute. Because querying slows down the processing of a policy, the virtual attribute values are normally cached for the user session.

  Change the value of this option from session to a more frequent interval only on those attributes that are critical to the security of your system or to the design of your work flow. You can select to cache the value for the session, for the request, or for a time interval varying from 5 seconds to 60 minutes. For more information, see "Using the Refresh Data Option" on page 692.

- **X-Forwarded-For IP:** Injects the X-Forwarded-For IP address of the client.
- **String Constant:** Injects a static value that you specify in the text box. This value is used by all users who access the resources assigned to this policy.
- **Data Extension:** (Conditional) If you have installed a data extension for Identity Injection policies, this option injects the value that the extension retrieves. For more information about creating a data extension, see NetIQ Access Manager Developer Tools and Examples.

The value type you use depends upon your application setup.

7 Specify values in the following fields:

Domain: Select one of the following:

- **LDAP Attribute:** When the user is authenticated at the Identity Serer by using a Kerberos authentication. This attribute uses userPrincipalName of the user from Active directory.
- **String Constant:** When the user is authenticated at the Identity Server by using a non-Kerberos authentication. If the required domain is not available in any LDAP attribute, the administrator can specify the domain name manually.

Target Host: Select from request: Selects the Web server FQDN that user has configured while configuring Web servers of the proxy service.

Multi-Value Separator: Select a value separator, if the value type you have select is multi-valued. For example, **Roles** can contain multiple values.

DN Format: If the value is a DN, select the format for the DN:

- **LDAP:** Specifies LDAP typed comma notation:

  ```
  cn=jsmith,ou=Sales,o=novell
  ```

- **NDAP Partial Dot Notation:** Specifies eDirectory™ typeless dot notation.

  ```
  jsmith.sales.novell
  ```

- **NDAP Leading Partial Dot Notation:** Specifies eDirectory typeless leading dot notation.

  ```
  .jsmith.sales.novell
  ```

- **NDAP Fully Qualified Partial Dot Notation:** Indicates eDirectory typed dot notation.

  ```
  cn=jsmith.ou=Sales.o=novell
  ```

- **NDAP Fully Qualified Leading Dot Notation:** Indicates eDirectory typed leading dot notation.

  ```
  .cn=jsmith.ou=Sales.o=novell
  ```

8 Click **OK**.

9 (Optional) To add a second rule, click **New** in the Rule List.

You can inject only one Kerberos ticket into an Identity Injection rule. However, your policy can have multiple rules. If you inject two Kerberos tickets, each in a separate rule, the Kerberos ticket in the rule with the highest priority is applied. The Kerberos ticket action in the second rule is ignored.

10 Click **OK** > **Apply Changes**.

Importing and Exporting Identity Injection Policies

You can import and export Identity Injection policies to run them in other Access Manager Appliance configurations. The policy is exported as a text file with XML tags.

NOTE: We do not recommend editing the exported file with a text editor. Any changes you want to make to a policy must to be done through the Administration Console.

To export an Identity Injection policy:

1 In the Administration Console, click **Policies > Policies**.

2 Select the policy container.

3 Select an Identity Injection policy, then click **Export**.

4 (Optional) Modify the name suggested for the file.

5 Click **OK**.

6 Using the features of your browser, specify where the file is to be copied.

To import a policy:

1 Ensure that any referenced shared secret stores have been created. See “Creating and Managing Shared Secrets” on page 732.

2 If the policy uses LDAP or Liberty Profile attributes, ensure that the Identity Server has been configured for these same attributes.

3 Ensure that any referenced role policies have been imported.

See “Importing and Exporting Role Policies” on page 642.

4 Click **Policies**.

5 Click **Import**, then browse to the location of the file.

6 Click **OK**.

7 When the policy appears in the list, click **Apply Changes**.

Sample Identity Injection Policy

One of the common uses of an Identity Injection policy is to differentiate between internal users and external users. Web servers that have been configured for this logic can then display one set of pages to internal users and another set of pages to external users. The following sample policy is based on an environment that has the following characteristics:

- The Web server has been configured to look for a custom tag called `IPAddress` and to differentiate between internal IP addresses and external IP addresses.
- The internal customers have NAT IP addresses.
- The protected resource is a page called `mycompany.html`. This page is a public protected resource (no authentication required) because the IP address of the client is available before authentication.

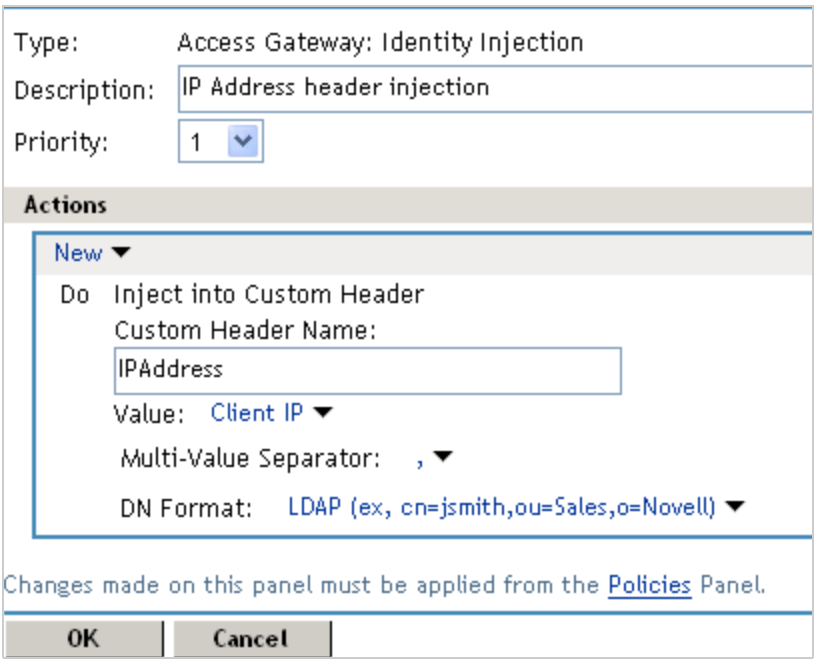
To configure your site for this type of policy:

1. In the Administration Console, click **Policies > Policies**.
2. Select the policy container.
3. Click **New**, specify a name for the policy, select **Access Gateway: Identity Injection** for the type, then click **OK**.
4. In the **Actions** section, click **New** > **Inject into Custom Header**.
5. Fill in the following fields:

   Custom Header Name: Specify `IPAddress` in the text box.

   Value: Select **Client IP**.

   The other fields do not need to be modified. Your policy must look similar to the following:



6. Click **OK** twice, then click **Apply Changes**.
7. Assign the policy to the `mycompany.html` page of the Web server. Click **Access Gateways** > **Edit** > **[Name of Reverse Proxy]** > **[Name of Proxy Service]** > **Protected Resources**.
8. In the Protected Resource List, select the protected resource for the page or click **New** to create one, then specify a name for it.
9. In the **URL Path List**, ensure that the path ends with the name of the page. For example:

   ```
   /mycompany.html
   ```

10. Click **Identity Injection**, select the name of the IP address policy, then click **Enable**.
11. To save the changes, click **Configuration Panel** > **OK**.
12. On the Configuration page, click **OK**, then click **Update**.
13. Configure the Web server to use the IPAddress values in the custom header to distinguish between external and internal customers.

    In this sample scenario, the Web server is configured to recognize IP addresses starting with `10.` as internal customers and all other addresses as external customers.

Form Fill Policies

A Form Fill policy allows you to pre-populate fields in a form on first login and then save the information in the completed form to a secret store for subsequent logins. The user is prompted to reenter the information only when something changes such as when a password expires. Form Fill is one of the features of Access Manager Appliance that enable you to provide single sign-on for your users.

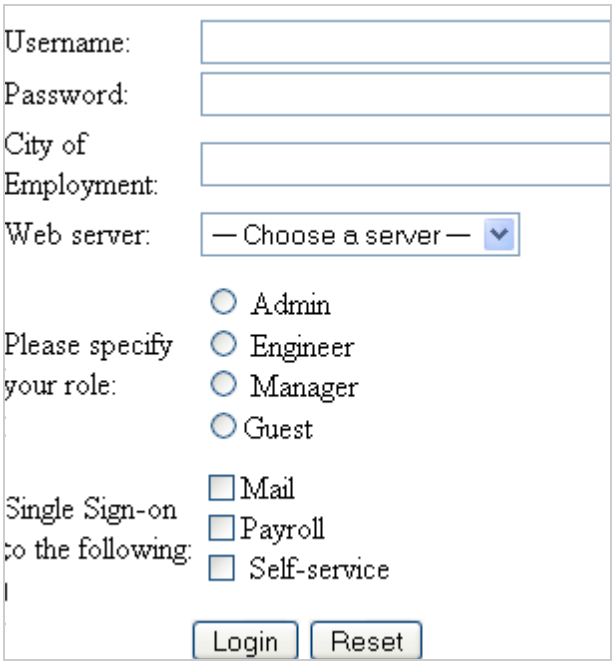
The HTML page determines the requirements for the Form Fill policy. This section describes the following:

- "Understanding an HTML Form" on page 711
- "Creating a Form Fill Policy for the Sample Form" on page 714
- "Implementing Form Fill Policies" on page 716
- "Creating and Managing Shared Secrets" on page 732
- "Importing and Exporting Form Fill Policies" on page 734
- "Configuring a Form Fill Policy for Forms With Scripts" on page 735

Understanding an HTML Form

The following figure is an example of a Web page containing an HTML form:

Figure 6-26 *Sample HTML Form*



The information in this section uses this sample form to explain how to create a policy. This sample form deliberately contains a variety of field types:

- Input items for Username and Password
- Selection options for the Web server field

- Radio buttons for the role
- Check boxes for single sign-on

When you analyze a form, you need to decide if you want the policy to fill in all the fields or just some of them. You then need to look at the source HTML of the form to discover the names of the fields and their types.

An HTML form is created using a set of HTML tags. A form consists of elements such as fields, menus, check boxes, radio buttons, and push buttons that control how the form is completed and submitted. For more detailed information about forms, see the Forms section at www.w3.org (http://www.w3.org/TR/html401/interact/forms.html).

The following HTML data corresponds to the sample form (see Figure 6-26). The lines that contain the information needed to create a Form Fill policy appear in bold type. Each line corresponds to a field in the form that requires information or allows the user to select information.

In the example, each bold line contains information about a field, its name, and type. You use this information in the policy to specify how the information in the field is filled.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
   "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
  <title>Form Fill Test Page</title>
</head>
<body>
  <form name="mylogin" action="validatepassword.php" method="post"
        id="mylogin">
    <table align="center" border="0" cellpadding="4" cellspacing="4">
      <tr align="center" valign="top">
        <td>
          <p align="center"><font size="5">Novell Services Login
             </font></p>
          <table align="center" border="0">

            <tr align="left">
              <td>Username:</td>
              <td><input type="text" name="username" size="30"></td>
            </tr>

            <tr align="left">
              <td>Password:</td>
              <td><input type="password" name="password" size="30">
              </td>
            </tr>

            <tr align="left">
              <td>City of<br>Employment:</td>
              <td><input type="text" name="city" size="30"></td>
            </tr>

            <tr align="left">
              <td>Web server:</td>
              <td>
                <select name="webserv" size="1">
                  <option value="default" selected>
                    --- Choose a server ---
                  </option>
                  <option value="Human Resources">
```

```
        Human Resources
      </option>
      <option value="Development">
        Development
      </option>
      <option value="Accounting">
        Accounting
      </option>
      <option value="Sales">
        Sales
      </option>
    </select>
  </td>
</tr>

<tr>
  <td colspan="2" align="left" height="25" valign="top">
    <p></p>
  </td>
</tr>

<tr align="left">
  <td>Please specify<br>your role:</td>
  <td>
     <input name="role" value="admin" type="radio">
            Admin<br>
     <input name="role" value="engineer" type="radio">
           Engineer<br>
     <input name="role" value="manager" type="radio">
           Manager<br>
     <input name="role" value="guest" type="radio">Guest
  </td>
</tr>

<tr>
  <td colspan="2" align="left" height="25" valign="top"
      width="121">
    <p></p>
  </td>
</tr>

<tr align="left">
  <td>Single Sign-on<br>to the following:</td>
  <td>
    <input name="mail" type="checkbox">Mail<br>
    <input name="payroll" type="checkbox">Payroll<br>
    <input name="selfservice" type="checkbox">
                Self-service<br>
  </td>
</tr>
```

```
          </table>
        </td>
      </tr>

      <tr>
        <td colspan="2" align="center">
          <input value="Login" type="submit">
          <input type="reset">
        </td>
      </tr>
    </table>
  </form>
</body>
</html>
```

Creating a Form Fill Policy for the Sample Form

The sample form has ten input fields and five selection options that need to be configured in the Form Fill policy. The following steps explain how to create a shared secret to store the values and use that shared secret to create a Form Fill policy for this sample form. For information about configuring the form fill policy for a complicated form with JavaScript, see “Configuring a Form Fill Policy for Forms With Scripts” on page 735.

1. In the Administration Console, click **Policies > Policies**.
2. Select the policy container, then click **New**.
3. Specify a display name for the policy and select **Access Gateway: Form Fill** for its type.
4. (Optional) Specify a description for the Form Fill policy. This is useful if you plan to create multiple Form Fill policies.

   You might want to specify the name of the HTML page that contains the form this policy is designed to fill.
5. In the **Actions** section, click **New**, then select **Form Fill**.
6. In the **Form Selection** section, select **Form Name** and specify **mylogin** in the text box. The form name comes from the HTML page. See the following line in the source for the page:

   ```
   <form name="mylogin" action="validatepassword.php" method="post"
   id="mylogin">
   ```
7. In the **Fill Options** section, specify all the input fields and select options. For each new field, click **New**. Specify the fields in the order in which they appear on the form. For items that are not available in the other data types such as an LDAP or Liberty attribute, create shared secrets to store the value.

   The following table displays the Fill Options selected for each input field:

Form Name	Fill Options
username	**Input Field Name:** username
	Input Field Type: Text
	Input Field Value: Credential Profile: LDAP Credentials: LDAP User Name
password	**Input Field Name:** password
	Input Field Type: Password
	Input Field Value: Credential Profile: LDAP Credentials: LDAP Password

Form Name	Fill Options
webserv	**Input Field Name:** webserv **Input Field Type:** Select **Input Field Value:** Shared Secret: sampleLogin: webserv To create this shared secret, click **New Shared Secret**, specify **sampleLogin**, then click **OK**. Select **sampleLogin**, click **New Shared Secret Entry**, specify webserv, then click **OK**. For more information, see "Creating and Managing Shared Secrets" on page 732. To add more entries to the same secret store, such as role and mail, you need to manage the secrets from the Identity Server. Save your draft of the policy, then click **Devices > Identity Servers > Shared Settings > Custom Attributes**. Select the name of your secret store (in this example it is sampleLogin). Add the entries you need for role, mail, payroll, and selfservice. These names need to match the form name.
role	**Input Field Name:** role **Input Field Type:** Radio Button **Input Field Value:** Shared Secret: sampleLogin: role
mail	**Input Field Name:** mail **Input Field Type:** Checkbox **Input Field Value:** Shared Secret: sampleLogin: mail
payroll	**Input Field Name:** payroll **Input Field Type:** Checkbox **Input Field Value:** Shared Secret: sampleLogin: payroll
selfservice	**Input Field Name:** selfservice **Input Field Type:** Checkbox **Input Field Value:** Shared Secret: sampleLogin: selfservice

8 In the **Submit Options** section, fill in the following fields:

Auto Submit: Select this option to submit the form as soon as all the values are filled in. If this option is not selected, even though all the values are filled in for the user, the user must click the **Submit** button.

Debug Mode: Select the **Debug Mode** option, which allows you to verify that the information is correct before submitting the form. If values must be filled in, you first see the form to add the values. When the form is submitted, you are presented with a JavaScript that contains all of the name/value pairs. To submit the form, you need to click the **Submit** button.

Insert Text in Header: Select this option so you can add a static value. In the **Text to Insert** box, specify the city value.

```
city = Provo
```

9 To create a login failure policy, click **New** in the **Actions** section, then select **Form Login Failure**.

10 In the **Form Selection** section, select **Form Name** and specify **mylogin** in the text box. The form name comes from the HTML page.

11 In the **Login Failure Processing** section, fill in the following field:

Clear Shared Secret Data Values from Policy: Select this option to clear the data stored in the Shared Secret object when login fails. Select the name you have given to this policy.

12 Use the up-arrow button to move the Form Login Failure policy to the top of the policy list.

You want the failure policy to execute first on login failure.

13 To create an Inject JavaScript policy, click **New** in the **Actions** section, then select **Inject JavaScript**. This option adds the configured JavaScript to a HTML page and is available only in interactive mode. For more information about creating an Inject JavaScript policy, see “Creating an Inject JavaScript Policy” on page 727.

14 In the **Configure Javascripts** section, select the option where you want the JavaScript inserted in the HTML page.

15 Click **OK**.

16 On the Policies page, click **Apply Changes**.

Implementing Form Fill Policies

“Creating a Form Fill Policy for the Sample Form” on page 714 section describes how to create a simple Form Fill policy for a few input fields. This section describes all available options and explains how to use them to create a Form Fill policy and a Login Failure policy.

- “Designing a Form Fill Policy” on page 716
- “Creating a Form Fill Policy” on page 721
- “Creating a Login Failure Policy” on page 727
- “Creating an Inject JavaScript Policy” on page 727
- “Troubleshooting a Form Fill Policy” on page 730

Designing a Form Fill Policy

Besides analyzing the form and determining the data items that need to be filled (see “Understanding an HTML Form” on page 711), you need to consider the following when designing the Form Fill policy:

- “Verifying the Content or Page Type of the Form” on page 716
- “Creating a Form Matching Rule” on page 717
- “Including JavaScript in a Form Fill Policy” on page 719
- “Form Fill Character Sets (UTF-8)” on page 721

Verifying the Content or Page Type of the Form

When configuring the protected resource that uses a Form Fill policy, the URL in the **URL Path List** must include the filename of the page that contains the form. Sometimes this is not possible. If the URL references a directory, the Access Gateway has to parse the files that match the URL and determine which one contains the form.

The Access Gateway Appliance checks the files for the following content types:

```
text/html
text/xml
text/css
text/javascript
application/javascript
application/x-javascript
```

If a file has no content type or has a type other than one in the above list, the Access Gateway Appliance skips the file.

The Access Gateway Service does not check for content type; it just parses the files that match the URL.

Creating a Form Matching Rule

To create a successful Form Fill policy, you need to create a matching rule that matches the policy to the HTML page that contains the form, and then matches the form on the page. The Access Gateway uses the following rules, in the order listed, when determining whether a page contains the required form:

1. Matches the protected resource path in the URL with the page. If they don't match, the page is rejected. If they match, continues. For more information, see "Using the URL of the Protected Resource" on page 717.
2. Checks for CGI criteria. If they don't match, the page is rejected. If they match or no criteria is specified, continues. For more information, see "Using CGI Matching Criteria" on page 718.
3. Checks for page matching criteria. If they don't match, the page is rejected. If they match or no page matching criteria is specified, continues. For more information, see "Using Page Matching Criteria" on page 718.
4. Checks the form name criteria (which can be the `<FORM>` name attribute, the `<FORM>` ID attribute, or a number). If it does not' match, the page is rejected. If it matches, the form is processed. For more information, see "Using Form Name Criteria" on page 719.

When the Access Gateway uses URL or CGI criteria, it can make a match early in the filling process. This allows the Access Gateway to fill the data from the Web server and send it, almost simultaneously, to the browser. However, if the Access Gateway is configured to use page matching criteria, the Access Gateway must retrieve the entire page from the Web server, process it, and then determine whether the page needs to fill a form. All this processing must be completed before the Access Gateway can send any data to the browser. Unless the page is quite small, users will clearly perceive the delay.

The form name matching criteria are not used for page matching. They are used to determine which form on the page is selected. Use the following methods to match the page and the form:

- Using the URL of the Protected Resource
- Using CGI Matching Criteria
- Using Page Matching Criteria
- Using Form Name Criteria

Using the URL of the Protected Resource

When you assign a Form Fill policy to a protected resource, we recommend that the URL specified in the **URL Path List** contain the filename of the page. Usually, such a URL is enough to match the HTML page for the form. However, when pages are dynamically generated, the same filename is

sometimes used to display different pages. Sometimes you can't specify the filename in the URL. When this is the case, you need to use either the **CGI Matching Criteria** or the **Page Matching Criteria** to create an accurate page matching rule.

Using CGI Matching Criteria

If the page for the URL changes with the CGI portion of the URL (the portion that follows the question mark (?) and also called the query string), you can enter the CGI value. For example, consider the following URL:

```
http://webaccess.novell.com/servlet/webacc?Action=User.logout
```

If this is your URL, you can enter `Action=User.logout` as the value in the text box for the **CGI Matching Criteria** option. If the page generated from this URL always contains the page you want to match, you do not need to add any additional page matching criteria.

Using Page Matching Criteria

If your URL of your protected resource has the following characteristics, you need to use page matching criteria:

- The URL does not contain any CGI data.
- The URL displays generated pages that vary in content. For example, if your form fill login page and the login failure page share the same URL, you need to use page matching criteria.

Page matching criteria are the most processing-intense form of matching and must be avoided if possible, but sometimes they are the only method available to identify the page with the correct form. For example, suppose you have a login failure page and login page that use the same URL, with no CGI data. You can use page matching criteria to ensure that the Access Gateway matches the Form Fill policies for login and for login failure to the correct pages. You need to examine the source code for each page, and identify a string at the top of the page that uniquely identifies the page.

For example, the login page might contain a `<TITLE>` element that names the application the user is logging in to. If the login failure page does not contain the same `<TITLE>` element, you can use the `<TITLE>` element to identity the login page. Suppose that this is true and the login page contains the following string:

```
<TITLE>Novell WebAccess</TITLE>
```

You would add this string as the value in the text box for the **Page Matching Criteria** option. Remember that white space is significant when white space is entered to the left of the value in the text box. To have the Access Gateway ignore white space, left-justify the value in the text box, or to ensure the correct amount of white space, copy and paste the HTML text directly from the source code of the Web page.

Now you need to uniquely identify the login failure page. If this page does not have a `<TITLE>` element, look at the strings near the top of the page. Suppose the page contains the following string:

```
"Please log in again. You might have typed your name or password incorrectly."
```

Because the login page does not contain this string, you can use this string to identify the login failure page. You would add the following string as the value in the text box for the **Page Matching Criteria** option for the login failure Form Fill policy.

```
Please log in again.
```

To have the Access Gateway ignore white space, left-justify the value in the text box, or to ensure the correct amount of white space, copy and paste the HTML text directly from the source code of the Web page.

Using Form Name Criteria

After identifying the page, the Access Gateway needs to identify the form on the page. If there is only one form on the HTML page, the Access Gateway can easily identify the form. If the form has a name or an ID attribute, you can use the value of the attribute to identify the form. If the form does not' have either of these attributes, you can use the **Number** option with a value of 1. The first form the Access Gateway finds on the page matches.

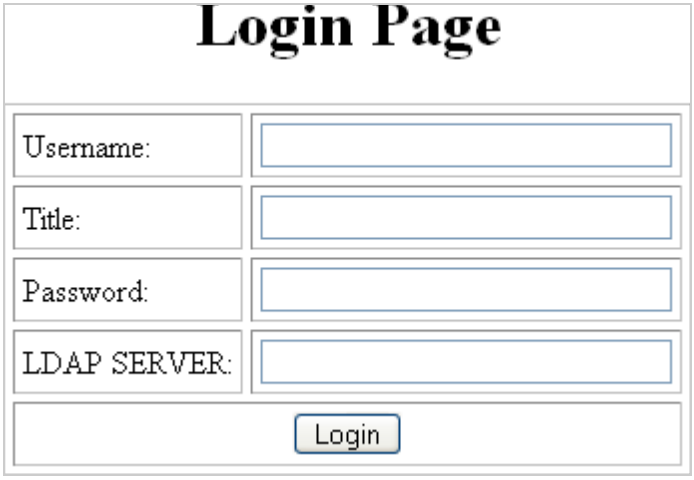
When multiple forms exist on the same HTML page, the easiest and fastest matching method is to give each form a unique name or unique ID on the HTML page. If the forms have the same name or ID, you need to use the Number option, and the order in which they appear on the page determines their number.

The value 0 for the **Number** option has special meaning. You use this value when you want the Form Fill policy to fill in values for all forms on the page. Sometimes a page has multiple forms, but all forms on the page must be filled in before the page can be submitted. For example, one form might contains user information and another form contain user preferences. If both of these forms need to be filled in before the user can log in, then you can use the Number option set to 0, and the Fill Options section of the policy can contain fields for both forms, in the order in which they appear on the page.

Including JavaScript in a Form Fill Policy

The following figure illustrates a simple form:

Figure 6-27 *Form Login Page*



The source code for this simple form reveals that it includes JavaScript functions:

```
<html><head><title>Login Page</title></head><body>
<h1 align="center">Login Page</h1>
<script language="JavaScript">
  function setCookie(){
    document.cookie="myCookieName=myCookieValue";
  }
  function validate(){
  if(document.mylogin.title.ldap.length == 0){
    alert("You must provide the title for the user!");
    return false;
  }
  return true;
}
</script>
```

```
<form name="jscript" action="viewInfo.php" method="post" onload="setCookie()">
<center>
<table border="1" cellpadding="4" cellspacing="4">
  <tbody><tr>
  <td>Username:</td>
  <td><input name="username" size="30" type="text"></td>
  </tr>

  <tr>
  <td>Title:</td>
   <td><input name="title" size="30" type="text"></td>
  </tr>
  <tr>
   <td>Password:</td>
  <td><input name="password" size="30" type="text"></td>
  </tr>

  <tr>
    <td>LDAP SERVER:</td>
  <td><input name="ldap" size="30" type="text"></td>
  </tr>
  <tr>
  <td colspan="2" align="center">
  <input value="Login" onclick="return validate();" type="submit">
  </td>

  </tr>
</tbody></table>
</center>
</form>

<script language="JavaScript">
function doCookie(){
document.cookie="myCookieName=myCookieValue";
}
return true;
}
</script>

</body></html>
```

The significant code snippets for determining whether to include JavaScript commands in the Form Fill policy are displayed in bold. The `<script>` elements are in bold because you need to be aware of all the JavaScript on the HTML page. Whether all the functions in the JavaScript need to be included in the policy is usually determined by trial and error. There are some clues you can use to determine the requirements:

- If a function is called within the form, you must include it in the Form Fill policy. The above form calls two JavaScript functions, `setCookie()` and `validate()`.
- If a function is not called by the form, you probably do not need to include it. The above form has one JavaScript function that falls within this category, `doCookie`. You can probably leave out these types of functions, but only trial and error can determine whether that is true.

For this form, select the **Auto Submit** option and the **Enable JavaScript Handling** option. If you wanted to test whether the `doCookie()` function was needed, you would specify the following in the **Functions to Keep** text box:

```
function setCookie()
function validate()
```

Each function needs to be placed on a separate line. This feature does a string compare, so the string after the function key word must match exactly a string in the JavaScript.

Form Fill Character Sets (UTF-8)

Access Manager Appliance supports only UTF-8 encoding (UCS Transformation Format 8) and ISO 8859-1. Otherwise, Form Fill translations to the secret data store cannot be guaranteed.

Creating a Form Fill Policy

1. Examine the source code for the HTML form and determine what data the form requires and where that data is stored (LDAP attributes, Liberty User Profile attributes, shared secrets, credential profiles, etc.)

   Ideally, the form must be its own HTML page, and the page must be as small as possible. Form Fill must parse the entire file and assemble the body in contiguous memory before the first byte of the form is displayed to the user. For a large file, this can take enough time that your users might think the system has a problem.

   If it isn't possible to have the form on its own HTML page, ensure that the form is easily identifiable on the page. For example, give the form a name or use CGI data (the text that the follows the question mark in the URL) to identify the page and form.

2. In the Administration Console, click **Policies > Policies**.
3. Select the policy container, then click **New**.
4. Specify a name for the policy, select **Access Gateway: Form Fill** as its **Type**, then click **OK**.
5. Fill in the following fields:

   Description: (Optional) Describe the purpose of this policy. Because Form Fill policies are customized to match the content of a specific HTML page, you might want to include the name of the page as part of the description.

   Priority: Determines the order in which a rule is applied in the policy, when the policy has multiple rules. Form Fill does not use this field.
6. In the **Actions** section, click **New** and select **Form Fill**.
7. In the **Form Selection** section, specify how the Access Gateway can identify the form on the page. Select one or more of the following methods. Be specific and use as few of the methods as possible. For information about how to use these options effectively, see "Creating a Form Matching Rule" on page 717.

   Form Name: Identities the form on the HTML page. Select one of the following:

   - **Form Name:** If the `<form>` element on your HTML page specifies a name attribute, select **Form Name** and specify the value of the name attribute in the text box. For example, suppose your form contains the following:

     ```
     <form name="mylogin" action="validatepassword.php" method="post"
     id="form1">
     ```

     For this form, you would specify *mylogin* in the text box.
   - **Form Number:** The Access Gateway numbers forms sequentially from the top of the HTML page. If your page has multiple forms, you can use **Form Number** option and specify the form's sequential location in the text box.
   - **Form ID:** If the `<form>` element on your HTML page specifies an id attribute, select **Form ID** and specify the value of the id attribute in the text box.

For example, if your form contains the following:

```
<form name="mylogin" action="validatepassword.php" method="post"
id="form1">
```

For this form, you would specify **form1** in the text box.

For more information, see "Using Form Name Criteria" on page 719.

CGI Matching Criteria: Allows the Access Gateway to evaluate the query string in the URL (the portion after the question mark) to differentiate pages that have the same URL. Consider the following URL:

```
http://webaccess.novell.com/servlet/webacc?Action=User.login
```

For this URL, enter the following string in the text box for **CGI Matching Criteria**:

```
Action=User.login
```

If possible, copy the text from the form and paste it into the **CGI Matching Criteria** text box.

For more information, see "Using CGI Matching Criteria" on page 718.

Page Matching Criteria: Causes the Access Gateway to search the HTML page for the specified text. If the specified text is found on the page, the page is a match for the policy. If it isn't found, the page is not a match for the policy and the policy is not applied. For example, suppose your HTML page has the following string within the `<FORM>` element:

```
<title>Form Fill Test Page</title>
```

If you enter this string in the **Page Matching Criteria** box, the Access Gateway searches the form for this string. If it finds the string, it knows it has a match.

White space is significant. If the text in the text box is left-justified, the text can be found anywhere on the HTML page. If the text contains leading white space, such as ten spaces, the text must be found with ten leading spaces. If possible, copy the text as it appears on the form and paste it into **Page Matching Criteria** text box.

The more specific your information is, the faster Access Gateway can match the form. Parsing page matching criteria is a very intensive process. If possible, use the URL path specified for the protected resource or **CGI Matching Criteria** to identity the form.For more information, see "Using Page Matching Criteria" on page 718.

8 In the **Fill Options** section, create an entry for all the input fields and select options in the form. For each input field or select option, you need to specify the following information:

Input Field Name: Specifies the name of the field or option. This is the name attribute or the ID attribute of the element on the form.

NOTE: If both name and ID attributes are available, always specify the name attribute in **Input Field Name**. If you specify the ID attribute, form fill does not work.

If only the ID attribute is available, the form gets filled with values, but auto submit does not work.

Input Field Type: Specifies the type attribute for the input field or select option in the form. Select one of the following data types for the field:

- **Text:** Indicates that the field is a text field on the form.
- **Password:** Indicates that the field is a password field on the form.
- **Checkbox:** Indicates that the field is a check box on the form.
- **Radio Button:** Indicates that the field is a radio button on the form.
- **Select:** Indicates that the field is a select option on the form.

- **Hidden:** Indicates that the field is an input field, but that this field is hidden from the user.
- **Not Specified:** Indicates that the field is an input field, but the data type is not specified in the form.

Input Field Value: Specify the value for the field. You must specify the data type, then enter the value. Select one of the following data types:

- **Credential Profile:** Specifies that the value must be retrieved from the credentials the user specified during authentication. If you have created a custom contract that uses credentials other than the ones listed below, do not use the Credential Profile as an input value.
  - **LDAP Credentials:** If you prompt the user for a username and password, select this option, then either **LDAP User Name** (the cn of the user) or **LDAP User DN** (the fully distinguished name of the user). Your Web server requirements determine which one you use.

    The default contracts assign the cn attribute to the Credential Profile. If your user store is an Active Directory server, the SAMAccountName attribute is used for the username and stored in the cn field of the LDAP Credential Profile.
  - **X509 Credentials:** If you prompt the user for a certificate, select this option, then select one of the following option depending on your Web server requirements.
    - **X509 Public Certificate Subject:** Specifies that the subject field from the certificate must be the value, which can match the DN of the user, depending upon who issued the certificate.
    - **X509 Public Certificate Issuer:** Specifies that the issuer field from the certificate must be the value, which is the name of the certificate authority (CA) that issued the certificate.
    - **X509 Public Certificate:** Specifies that the entire certificate must be the value.
    - **X509 Serial Number:** Specifies that the certificate serial number must be the value.
  - **SAML Credential:** Injects the SAML assertion as the value of the field when SAML is used for authentication. This value is usually used for the user's password.
- **LDAP Attribute:** Indicates that the value must be retrieved from the specified LDAP attribute. If the attribute you require does not appear in the list, click **New LDAP Attribute** to add the attribute.

  The **Refresh Data Every** option allows you to determine when to send a query to the LDAP server to verify the current value of the attribute. Because querying the LDAP server slows down the processing of a policy, LDAP attribute values are normally cached for the user session.

  Change the value of this option from session to a more frequent interval only on those attributes that are critical to the security of your system or to the design of your work flow. You can select to cache the value for the session, for the request, or for a time interval varying from 5 seconds to 60 minutes.
- **Liberty User Profile:** Indicates that the input field contains a Liberty User Profile attribute. In the value field, select the attribute. The attribute you select must be mapped to an LDAP attribute, and the Access Gateway retrieves its value from the LDAP directory.
- **Shared Secret:** Indicates that the input field contains a user-entered value that is to be stored in the specified shared secret store.

  You can create your own value. Click **New Shared Secret**, specify a display name for the store, and Access Manager Appliance creates the store. Select the store, click **New Shared Secret Entry**, specify a name for the attribute, then click **OK**. The store can contain one name/value pair or a collection of name/value pairs. For more information, see "Creating and Managing Shared Secrets" on page 732.

NOTE: To store user-entered value in Shared Secret, ensure that you have specified the name attribute in **Input Field Name**. The ID attribute does not work with Shared Secret.

The **Refresh Data Every** option allows you to determine when to send a query to verify the current value of the secret. Because querying slows down the processing of a policy, secret values are normally cached for the user session.

Change the value of this option from session to a more frequent interval only on those secrets that are critical to the security of your system or to the design of your work flow. You can select to cache the value for the session, for the request, or for a time interval varying from 5 seconds to 60 minutes.

- **Virtual Attribute:** Indicates that the value must be retrieved from the specified virtual attribute.

  The **Refresh Data Every** option allows you to determine when to send a query to verify the current value of the virtual attribute. Because querying slows down the processing of a policy, the virtual attribute values are normally cached for the user session.

  Change the value of this option from session to a more frequent interval only on those attributes that are critical to the security of your system or to the design of your work flow. You can select to cache the value for the session, for the request, or for a time interval varying from 5 seconds to 60 minutes. For more information, see "Using the Refresh Data Option" on page 692.

- **String Constant:** Indicates that the input field contains a static value. In the text box, specify the value for the string constant.
- **Data Extension:** (Conditional) If you have installed a data extension for Form Fill policies, injects the value that the extension retrieves. For more information about creating a data extension, see NetIQ Access Manager Developer Tools and Examples.

NOTE: To improve the policy's performance, configure the LDAP Attributes, Credential Profile, Liberty User Profile, and Shared Secret attributes to be sent with authentication. For more information, see "Configuring the Attributes Sent with Authentication" on page 156.

Data Conversion: Specify whether the case of the value entered by the user must be converted. Select one of the following options:

- **None:** Indicates that no conversion must be performed on the value.
- **To Upper Case:** Indicates that the value must be converted to uppercase.
- **To Lower Case:** Indicates that the value must be converted to lowercase.
- **LDAP DN to NDAP Partial Dot Notation:** Converts the LDAP DN (which uses typed comma notation) to eDirectory™ typeless dot notation.

  ```
  cn=jsmith,ou=Sales,o=novell to jsmith.sales.novell
  ```

- **LDAP DN to NDAP Leading Partial Dot Notation:** Converts the LDAP DN to eDirectory typeless leading dot notation.

  ```
  cn=jsmith,ou=Sales,o=novell to .jsmith.sales.novell
  ```

- **LDAP DN to NDAP Fully Qualified Partial Dot Notation:** Converts the LDAP DN to eDirectory typed dot notation.

  ```
  cn=jsmith,ou=Sales,o=novell to cn=jsmith.ou=Sales.o=novell
  ```

- **NDAP Fully Qualified Leading Dot Notation:** Indicates eDirectory typed leading dot notation.

  ```
  .cn=jsmith.ou=Sales.o=novell
  ```

Shared Secret Type: This option allows you to choose how the value you specified in the HTML form must be stored in the shared secret store.

- **None:** When you select this default option, the value that you specified in the HTML form will be stored and retrieved from the shared secret store on subsequent login.
- **Remember:** This option allows you to only store the value that you specified in the HTML form to the shared secret store.
- **Fill:** This option allows you to only retrieve the value from the shared secret store.

Example 6-1 *Configuring a Form Fill Policy to Change Password Using Different Shared Secret Types*

For example, if you want to change password on a HTML form, the form will have **Old Password**, **New Password**, and **Confirm Password** fields. While configuring the Form Fill policy, on the password change page, select the shared secret type as **Fill** for the **Old Password** field. For the **New Password** and **Confirm Password** fields select shared secret type as **Remember**. All the three input field names must point to the same shared secret entry.

When you access the password change page, the old password will be auto filled. The **New Password** and **Confirm Password** fields will be blank. Enter the **New Password** and **Confirm Password** fields and submit the page. The **Old Password** will be replaced with the **New Password** in the shared secret entry.

9 In the **Submit Options** section, specify how you want the information in the form submitted to the Web server. (The HTML form page determines whether the post method or the get method is used for the submission.) Select one or more of the following options:

Auto Submit: Indicates that you want the form submitted to the Web server without having the user confirm the submission by clicking a **Submit** button. If this option is not selected, Form Fill can fill in the data, but the user must click the **Submit** button before the data is sent to the Web server. When the form is not auto submitted, all the JavaScript on the form is executed.

If you select **Auto Submit**, you can select one or more of the following options:

- **Debug Mode:** Allows you to verify that the information in the filled-in form is valid before it is posted to the Web server. You can right-click and view the source that is being submitted to the Web server. If it is correct, click **Submit** to send it to the Web server.

  This is a troubleshooting option. We recommend that you use it when creating a new Form Fill policy, and that you remove it when you have determined that the policy is behaving as expected.
- **Mask Data:** Replaces text input field values (username, password, etc.) with nov-ss-ff- masked instead of the value specified by the value parameter when the form is sent to the browser. The Access Gateway replaces these masked values with the real values when the Access Gateway submits the form to the Web server. The user's browser never sees the actual values for these fields.
- **Detect Loop:** In some scenarios, Form Fill processor tries to auto submit the form and every time login fails, the form fill request goes into infinite loop. The Login Failure policy cannot handle the following scenarios:
  - When a Web server returns the same login page to the Access Gateway after login failure.
  - When a Web server uses URL redirection or forwarding method to redirect a user to the same login page after login failure.

If you have selected **Auto Submit**, you can select the **Detect Loop** option. This option allows you to detect the loop and auto submit stops. Access Manager will now ask you to fill the form in an interactive mode.

This is achieved by creating a cookie in the browser, which will calculate the number of times the same form is posted to the Access Gateway in a given period of time. These values are set to 3 submits in 6 seconds.

Limitations:

- Use these options only when the Login Failure policy cannot detect or handle looping.
- When Web server returns a different login form depending on a query-string or CGI portion of the URL, these options may not work as expected.

Insert Text in Header: If this option is selected, you can use the **Text to Insert** option to specify text to add to the header. Use this option to insert static values into the form.

Enable JavaScript Handling: Retains JavaScript from the original page if you have also selected the **Auto Submit** option. For a new Form Fill policy, you must also select the **Debug Mode** option so you can verify that you have included all the functions and statements that need to be executed in the policy.

Use the following fields to specify how you want the JavaScript handled:

- **Functions to Keep:** Specifies the functions you want executed from the JavaScript on the original page. By default, no functions on the page are executed. In the text box, use the following format:

  ```
  function setCookie()
  ```

  where `function` is a key word, followed by a space, and then the name of the function. Each function must be entered on a separate line, but you need only one function per script block. Everything must match exactly (name, capitalization, white space.) If you include the parentheses after the function name (setCookie()), they must exactly match the white space in the JavaScript. If possible, copy the function from the HTML page.

- **Statements to Execute on Submit:** Specifies the functions you want executed just before the form is posted. Copy the JavaScript statement from the HTML page or add a JavaScript statement that you want called that is not on the HTML page. This allows you to modify the behavior of the form when you can't modify the form.

  If the text box is empty, the JavaScript function specified in the submit field of the HTML page executes before the form is posted.

For more information, see "Including JavaScript in a Form Fill Policy" on page 719.

10 In the **Error Handling** section, specify how you want errors handled.

Redirect to URL: When an LDAP or NSS error occurs, the user is redirected to the URL you specify in the text box. This is optional and allows you to customize the error handling process. If you do not customize it, a standard error page is displayed.

11 Click **OK**, then click **Apply Changes**.

12 Continue with "Creating a Login Failure Policy" on page 727 or "Assigning a Form Fill Policy to a Protected Resource" on page 108.

Creating a Login Failure Policy

The Login Failure policy can be part of the same policy as the Form Fill policy, if both share the same URL. In this case, the Form Login Failure policy must be the first action in the policy, and the Form Fill policy must be the second action in the policy. This causes a login failure to execute the policy that clears the stored data and the Form Fill policy to prompt the user for new data.

If the user is redirected to a different page when login fails, it is best to create a separate policy for that page, create a protected resource that includes just that page, and assign your Form Login Failure policy to that resource.

To create a Login Failure policy:

1. In the Administration Console, click **Policies > Policies**.
2. Select the policy container, then click **New**.
3. Specify a name for the policy, select **Access Gateway: Form Fill** as its **Type**, then click **OK**.
4. In the **Actions** section, click **New** > **Form Login Failure**.
5. In the **Form Selection** section, identify the form. This section uses the same criteria for identifying a form as the Form Fill policy. For more information, see Step 7 on page 721 and “Creating a Form Matching Rule” on page 717.
6. In the **Login Failure Processing** section, define the actions you want executed when a user fails to log in. Fill in the following fields:

   Redirect to URL: When a user’s login attempt fails, use this option with its text box to specify the URL you want the user redirected to. This is optional and allows you to customize what happens on login failures.

   Clear Shared Secret Data Values From Policy: Select this field to delete the user’s stored data for a Form Fill policy. If the user has the ability (and perhaps the requirement) to periodically change his or her password or any other information about the form, you need to select this field. Otherwise, the wrong data can be stored for the user, and the Access Gateway has no way of updating the information.

   From the list of Form Fill policies, select the policy whose stored values must be cleared with this Login Failure policy.
7. Click **OK** > **Apply Changes**.
8. Continue with “Assigning a Form Fill Policy to a Protected Resource” on page 108.

Creating an Inject JavaScript Policy

The Inject JavaScript policy adds the configured JavaScript to a protected resource page, when used in the interactive mode. You can create a standalone Inject JavaScript policy. You can also use this policy with the Form Fill policy. When you use the Form Fill policy with this one, configure the actions in the following sequence:

1. Form Login Failure policy
2. Form Fill policy
3. Inject JavaScript policy

When the Inject JavaScript policy is configured along with the Form Fill policy, ensure that **Auto Submit** is not enabled for the Form Fill policy. In the **Configure Javascripts** section, select the option where you want to insert JavaScript in the HTML page. The following are examples based on the option you have selected.

In the head block

Selecting this option inserts the following JavaScript in the header:

```
<html>
<head>
<script language="JavaScript">
alert("Head");
</script>
```

At the beginning of the body block

Selecting this option inserts the following JavaScript just after the <body> tag.

```
<title> Test Java Script</title>
</head>
<body>
<script language="JavaScript">
alert("Begin Body");
</script>
```

At the end of the body block

Selecting this option inserts the following JavaScript just before the </body> tag.

```
BODY starts. <br>
Inject Java Script. <br>
BODY ends. <br>
<script language="JavaScript">
alert("End Body");
</script>
</body>
</html>
```

To create an Inject JavaScript policy:

1. In the Administration Console, click **Policies** > **Policies**.
2. Select the policy container, then click **New**.
3. Specify a name for the policy, select **Access Gateway: Form Fill** as its **Type**, then click **OK**.
4. In the **Actions** section, click **New** > **Inject JavaScript**.
5. In the **Form Selection** section, select the criteria. If you are creating a standalone Inject JavaScript policy, specify the criteria. For more information, see "Using CGI Matching Criteria" on page 718 and "Using Page Matching Criteria" on page 718. When you use the Inject JavaScript policy with the Form Fill policy, it uses the same criteria as that of the Form Fill policy.

   NOTE: If you do not specify a criteria in the **Form Selection** section, the Inject JavaScript policy is applied to all protected resource pages.
6. Click **OK** > **Apply Changes**.
7. Continue with "Assigning a Form Fill Policy to a Protected Resource" on page 108.

Sample Inject JavaScript Policy

The script in this example assumes that the contract timeout is five minutes. After four minutes, the user gets a popup message (timeout.html) with an option to refresh the page. If the user clicks on the option, the session is extended. If the user closes the popup or does not respond to the message, the system executes AGLogout and the session gets terminated.

Perform the following steps:

1 Configure a Form Fill policy to insert Java Script into the head block of an HTML page.

1a Go to **Policies** > **New**.

1b Specify a name for the policy and select the type as **Access Gateway: Form Fill**.

1c In the **Actions** section, click **New** and then select **Inject JavaScript**.

1d Define a **CGI matching Criteria** or **Page Matching Criteria**.

1e Under the **Configure Javascript** section, select **In Head Block**.

1f Click **Configure JavaScript** and copy the following script:

```
<script language="JavaScript">
var x;
var timerID ;

function timeoutClock()
{
if(x==60) // 60 seconds is the session time left when the session expire
warning message appears.
{
newwindow =
window.open('timeout.html','toWindow','toolbar=no,menubar=no,resizable=no,
scrollbars=no,status=no,location=no,width=300,height=200');
}
if(x==0)
{
window.location.href = 'https://www.ag1.com:443/AGLogout' // AGLogout link.
}
x=x-1;

var t=setTimeout(function() {timeoutClock()} ,1000);

}
function resetClock()
{
clearTimeout(timerID);
x = 300; //5 Minutes. This is the contact timeout defined.
timeoutClock();
}
</script>
```

In this script, set the value of x inside the function resetClock() according to the contract time out. In the example, x is set 300 that is equivalent to five minutes. Also, modify the following link in the script based on your configuration. This is a simple AGLogout link.

```
window.location.href = 'https://www.ag1.com:443/AGLogout'
```

1g Click **OK** > **Apply Changes**.

2 Assign the policy to a protected resource.

For more information, see “Assigning a Form Fill Policy to a Protected Resource” on page 108.

3 Ensure to call the resetClock() function in the body tag of the HTML page (`<body onload=resetClock();>`). This initializes the counter to 300 every time the page is loaded.

4 Create a `timeout.html` page, which contains warning message for the user that the session is going to end soon. The content of `timeout.html` can be as follows:

```
<script type="text/javascript">
var howLong = 60000;
function closeMe()
{
var t = setTimeout(function() {self.close()},howLong);
}
function closeCurrentWindow()
{
  window.close();
}
</script>
Click <a href="javascript:window.opener.location.href =
window.opener.location.href;window.close()";>[here]</a>to refresh now.
<br>Else you will be logged out in 60 seconds
<body onload=closeMe();
Timeout
</body>
```

Troubleshooting a Form Fill Policy

When a new Form Fill policy is not behaving as expected, use the following tips to discover the cause:

- Select the **Debug Mode** option. This option prepares the form for submission, but does not'submit the form until you click the **Submit** button. This allows you to view the source, and determine if the policy is generating the required data.
- Ensure that all input fields have valid names, that the fields are being filled in the correct order, and that any JavaScript commands have been entered correctly.
- Enable Form Fill logging. Form Fill is a function of both the proxy service and the Embedded Service Provider. The Embedded Service Provider logs the evaluation of the policy, and the proxy logs the process of gathering the data. To enable the Embedded Service Provider tracing, see "Turning on Logging for Policy Evaluation" on page 871. To enable Access Gateway log entries for Form Fill policies, see "Enabling Form Fill Logging" on page 1041.

Check for the following problems with the source content of the Form Fill page:

- "Valid HTML Structure" on page 730
- "The Option Element Does Not Contain a Value Attribute" on page 731
- "The Form Element Does Not Contain a Method Attribute" on page 731

Valid HTML Structure

The Form Fill process aborts if the page does not contain valid HTML structure. The page must contain the `<html></html>` tags, and the form must contain the `<form></form>` tags. If these tags are missing, you must correct the source page on the Web server. If this is not possible, you can create a rewriter policy to add the tags.

- To add the `<html>` tag, have the rewriter policy search for the `<body>` tag, and replace it with `<html><body>`.

- To add the </html> tag, have the rewriter policy search for the </body> tag, and replace it with </body></html>.
- Use similar entries to add the <form></form> tags. You'll need to discover which tag or phrase starts and stops the form.

Configure your rewriter policy so that it runs before the default rewriter policy. For more information about rewriter policies, see "Configuring HTML Rewriting" on page 113.

The Option Element Does Not Contain a Value Attribute

If an <option> element does not contain a value attribute, Form Fill cannot fill the value. For example:

```
<form action="select.htm">
   <select name="top2">
      <option>Bob</option>
      <option>Alice</option>
   </select>
</form>
```

If your form contains <option> elements similar to these, they need to rewritten to contain a value attribute. For example:

```
<form action="select.htm">
   <select name="top2">
      <option value="name1">Bob</option>
      <option value="name2">Alice</option>
   </select>
</form>
```

If possible, change the source page on the Web server to add the value attribute to the <option> elements. If this is not possible, you can use a rewriter policy to add the value attribute.

- For the Bob option, have the rewriter policy search for <option>Bob and replace it with <option value="name1">Bob.
- For the Alice option, have the rewriter policy search for <option>Alice and replace it with <option value="name1">Alice.

Configure your rewriter policy so that it runs before the default rewriter policy. For more information about rewriter policies, see "Configuring HTML Rewriting" on page 113.

The Form Element Does Not Contain a Method Attribute

If the <form> element does not contain a method attribute, Form Fill does not run an Auto Post. For example, the following form cannot use an Auto Post.

```
<form name="loginForm">
```

To enable Form Fill so that it can run an Auto Post, you need to add a method attribute to the <form> element. For example:

```
<form method="get" action="index.htm" name="loginForm">
```

If possible, change the source page on the Web server to add the method attribute to the <form> element. If this is not possible, you can use a rewriter policy to add the method attribute.

- Search for <form
- Replace this string with <form method="get" action="index.htm"

Configure your rewriter policy so that it runs before the default rewriter policy. For more information about rewriter policies, see “Configuring HTML Rewriting” on page 113.

Creating and Managing Shared Secrets

A shared secret is an object that holds name and value pairs for Form Fill and Identity Injection policies.

- If your HTML form prompts the user for more than credential information, you need to create a shared secret to store the values.
- If your Web server requires some name/value pairs to be injected and these are not available from the HTTP request, you need to create a shared secret to store these name/value pairs so that they can be injected into the header before it is sent to the Web server.

Access Manager Appliance supports the creation and use of secrets from the following locations:

- In the local configuration store
- In eDirectory user stores that are running Novell SecretStore
- In a user store that has been configured with a custom attribute for secrets

---

NOTE: Before using Access Manager to store and encrypt secrets, ensure that you choose your **Preferred Encryption Method** and change the default **Encryption Password Hash Key** value. If either of these options is changed after any secrets are stored, Access Manager will not be able to retrieve the secrets.

For more information about configuring Access Manager Appliance to store secrets, see “Configuring a User Store for Secrets” on page 276.

This section describes the following topics:

- “Naming Conventions for Shared Secrets” on page 732
- “Creating a Shared Secret Independent of a Policy” on page 733
- “Modifying and Deleting a Shared Secret” on page 734

Naming Conventions for Shared Secrets

The policy engine allows you to create shared secrets and name the attributes for the store as you are creating an Identity Injection or Form Fill policy. When you create the shared secret, we recommend that you name the shared secret after the application for which you are creating the policy. Each value requires a name, and we recommend that you use the same name for the value name as the Input Field Name on a Form Fill policy or for the header name on an Identity Injection policy. For example if your e-mail application requires the e-mail address for the name on the login form, you could set up the following Shared Secret values:

Input Field Name	Input Field Value	Shared Secret Name	Entry Name
emailaddress	Shared Secret	emailapp	emailaddress

Your applications, how you use them, and your personal preferences determine whether you create one shared secret and use it for all your applications or whether you create a shared secret for each application.

- If the applications use some of the same secrets, you can use the same shared secret for these applications. In this case, give the shared secret a name that reflects all of the applications using it.
- If an application does not use the same secrets as another application and you want the freedom to remove the application and its secrets without affecting other applications, you must create a separate shared secret for this application.
- If you are using Novell SecretStore, the secret names specified in your Access Manager Appliance policies need to match the names you have already configured.

A local shared secret store does not contain any name/value pairs until you configure a Form Fill policy to add name/value pairs or enable the **Allow End Users to See Credential Profile** option. This option allows the username and password to be stored in the local secret store. To set this option, click **Devices > Identity Servers** > *Edit* > **Liberty** > **Web Service Providers** > **Credential Profile.**

NOTE: You can create/edit/delete the values of a shared secret in the following scenarios:

- When you use a Form Fill policy.
- When you login to Identity Server and use the default landing page. Click on **Profile** > **My Profile** > **Credentials** > **Credential List**. This will be allowed only after enabling the **Allow End Users to See Credential Profile** as mentioned above.
- When you use other NetIQ products such as Identity Manager and Secure Login. This can be used if you are using external eDirectory secret store.
- The Identity Injection policy can use the shared secrets, but will not allow to create/edit/delete the values of shared secrets.

Creating a Shared Secret Independent of a Policy

You can create a shared secret as part of the process of creating a Form Fill or Identity Injection policy. You can also create a shared secret independent of a policy:

1. In the Administration Console, click **Devices > Identity Servers**, then click **Shared Settings** > **Custom Attributes**.
2. To create a new shared secret, click **New** in the **Shared Secret Names** section, and fill in the following fields:

   Secret Name: Specify a display name for the shared secret.

   Secret Entry Name. Specify an attribute name for a value you want to store.
3. Click **OK**.

   The Identity Server creates and encrypts the object.
4. To create additional attributes to store values, click the secret name, click **New**, specify a name, then click **OK**.
5. Click **OK**.

Modifying and Deleting a Shared Secret

Before deleting a shared secret, you need to delete the policies that are using the shared secret or modify the policies to use a different shared secret. For information about deleting policies, see “Deleting Policies” on page 594.

Both Form Fill and Identity Injection policies can use shared secrets. The following instructions explain how to modify an Identity Injection policy to use a new shared secret and then how to delete the old shared secret.

1. In the Administration Console, click **Policies** > **Policies** > **[Name of Policy] > [Rule]**.
2. Select the **Value** field that uses the shared secret you want to delete. Click its name, then click **New Shared Secret**.
3. Specify the name for a new shared secret, then click **OK**.
4. Click the name of the shared secret, select the new shared secret store, then click **New Shared Secret Entry**.
5. Specify the attribute name for this shared secret entry, then click **OK**.
6. Modify any other **Value** fields to use the new shared secret. Create new attributes as needed.
7. To save the modifications to the policy, click **OK** twice, then **Apply Changes**.
8. To delete the old shared secret, click **Identity Servers** > **Shared Settings** > **Custom Attributes**.
9. Select the name of the old shared secret and the attributes, then click **Delete**.

Importing and Exporting Form Fill Policies

You can import and export the Form Fill policies in order to run them in other Access Manager Appliance configurations and to analyze the policy. The policy is exported as a text file with XML tags. We do not recommend editing the exported file with a text editor. Any changes you want to make to a policy ought to be done through the Administration Console.

To export a Form Fill policy:

1. In the Administration Console, click **Policies > Policies**.
2. Select a Form Fill policy, then click **Export**.
3. (Optional) Modify the name suggested for the file.
4. Click **OK**.
5. Using the features of your browser, specify where the file is be copied.

To import a policy:

1. Ensure that any referenced shared secret stores have been created. See “Creating and Managing Shared Secrets” on page 732.
2. If the policy uses LDAP or Liberty Profile attributes, ensure that the Identity Server has been configured for these same attributes.
3. In the Administration Console, click **Policies > Policies**.
4. Click **Import**, then browse to the location of the file.
5. Click **OK**.
6. When the policy appears in the list, click **Apply Changes**.

Configuring a Form Fill Policy for Forms With Scripts

The Form Fill policy created for the Access Gateway works well with forms that contain a **Submit** button whose `onclick` action submits the form data to the Web server without executing any JavaScript or VBScript. However, when HTML forms contain complicated JavaScript or VBScript, Form Fill for that form fails.

For example, single sign-on by using the Form Fill policy to fill and autosubmit a form fails if the Submit button or the login button requires execution of a JavaScript function before submitting the form data to the Web server.

The following sections explain why Form Fill fails with the Form Fill policy when the HTML form contains complicated JavaScript. This section also describes the procedure to configure a Form Fill policy for such forms.

- "Why Does Form Fill Fail with the Default Policy?" on page 735
- "Understanding How a Form Is Submitted" on page 737
- "Creating a Form Fill Policy for Autosubmission" on page 738
- "Configuring the Advanced Options for Autosubmission" on page 739

Why Does Form Fill Fail with the Default Policy?

The following section explains the process that takes place when a client requests a form that is configured with the Form Fill policy as described in Chapter , "Form Fill Policies," on page 711.

Figure 6-28 *Sample Login Form with JavaScript*



When the Access Gateway is configured with the default Form Fill policy, it adds the following function to the Login page received from the Web server. The bold text indicates where JavaScript is called.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<!--Generated by Apache Software Foundation (Xalan XSLTC)-->
<html class="detail/detail">

<head>
     <META http-equiv="Content-Type" content="text/html; charset=UTF-8">
          <script type="text/JavaScript">
             /*SET up global vars*/
            //all the variable declaration

         <script type="text/JavaScript">             function dvdRegisterSelect()
{                    }          </script>

     <title>Login Page</title>

</head>

<body id="tpz_body" style="width:99%; "onload="tpzOnLoad('login.prompt.g');
window.status='login.prompt.g'; ContextMenu.setup({'showForms':true});
ContextMenu.attach('detail/detail', 'cwc_optionsMenu_detail')"
onfocus="window.status='login.prompt.g'" >

     <script type="text/JavaScript">         var arReenable = new Array();
function enableAll() {return reenableControls(arReenable);}        </script>

     <script type="text/JavaScript">        //all the variable declaration
function verify( f, bSubmitToSelf ){ return verifyFields
(bSubmitToSelf,"\n");}      </script>

   <div>
      <a title="Login" class="tabSelected">Login</a>
   </div>
 <formname="topaz"id="topaz"method="post"action="detail.do"
onsubmit="enableAll();return verify(this,true);">
<input type="hidden" name="focus" id="focus" value="var/user.id">
<input type="hidden" name="focusContents" id="focusContents" VALUE="testuser1" >
<input type="hidden" name="focusId" id="focusId" VALUE="X2" >
<input type="hidden" name="formname" id="formname" VALUE="login.prompt.g">
<input type="hidden" id="clientWidth" name="clientWidth" VALUE="1473" >

     <script type="text/javascript">         function
printThisView(){tpzPrintDetail();}      </script>

<input type="text" id="X2" name="var/user.id" dvdVar=""
onclick="handleOnClick(this,event);"  VALUE="testuser1" scripttype="text">
<input type="password" id="X5" name="var/old.password" dvdVar=""
onclick="handleOnClick(this,
event);" " VALUE="novell081" >

<input type="button" name="0" id="X8" ButtonID="0" title="Login Page" value="Login"
onclick="tpzDrillTable('', 'Login', '0','listdetail')" >
<input type="button" name="3" id="X9" ButtonID="3" title="Exit Login Page"
value="Cancel" onclick="tpzDrillTable('', 'Cancel', '3','listdetail')" >
</form>

       <script language="JavaScript">           <!--              function
LAGSubmitForm()               {          document.forms[0].submit();
}             LAGSubmitForm();           //-->        </script>
</body>
</html>
```

In the above code, the `LAGSubmitForm()` function calls the default submit action of the form, which uses a POST request to send the data to the Web server. But the `submit` action for the sample login form requires a JavaScript function to be executed. This function in turn submits the form data to the Web server. However, because the JavaScript is not executed by the default Form Fill policy, posting of the form data fails:

```
row=&__x=&thread=0&event=0&transaction=0&type=detail&focus=var%2Fuser.id&focusCont
ents=testuser1&focusId=X2&focusReadOnly=&start=&count=&more=&tablename=&window=&cl
ose=&_blankFields=&_uncheckedBoxes=&formchanged=&formname=login.prompt.g&_multiSel
ection=&_multiSelection_tableId=&clientWidth=1473&var%2Fuser.id=testuser1&var%2Fol
d.password=novell081&var%2FL.language=en&0=Login&3=Cancel
```

Meanwhile, the browser expects to receive the following POST request and does not autosubmit the form:

```
row=&__x=&thread=0&event=0&transaction=0&type=detail&focus=var%2Fuser.id&focusCont
ents=testuser1&focusId=X2&focusReadOnly=null&start=&count=&more=&tablename=&window
=&close=&_blankFields=&_uncheckedBoxes=&formchanged=&formname=login.prompt.g&_mult
iSelection=&_multiSelection_tableId=&clientWidth=1217&var%2Fuser.id=testuser1&var%
2Fold.password=novell081&var%2FL.language=en
```

Note the difference in POST requests sent to the browser. The first POST request has `&0=Login&3=Cancel` appended, which causes the login to fail.

For the browser to send the proper POST data, the Access Gateway must add the following JavaScript statement to the **Statements to execute** section:

```
tpzDrillTable('','Login','0','listdetail');
```

Understanding How a Form Is Submitted

For the Access Gateway Appliance, you can configure the Form Fill policy to submit the form in the following ways:

- **Manual Submit:** When a form is configured for manual submission, all the fields configured in the Form Fill policy are automatically filled by the Access Gateway for the user. The user must then manually click the **Submit** button in the form to submit the form to the Web server protected by Access Gateway.
- **Autosubmit:** When Autosubmit is configured, the actual form is processed in such a way that all additional scripts not required to submit the form data to the Web server are removed. A temporary form is created on runtime with necessary form data in hidden format and with an additional `LAGSubmitForm()` function as follows:

  ```
  function LAGSubmitForm()
  {
  executeJavaScript();
  }
  LAGSubmitForm();
  ```

  In this example, `executeJavaScript()` is the function that executes the JavaScript or the VBScript statements configured in the **Statements to execute** section. If statements to be executed are present, you can also find the function definition for `executeJavaScript()` as follows:

  ```
  executeJavaScript()
  {
  document.forms[0].submit();
  }
  ```

In this example, `form[0]` is the single form in the HTML page and `submit` is the default action associated with the submit or login button of the form that automatically submits the form to the Web server. This approach works for forms where the default action of the **Submit** button is to submit a POST request for the form data.

- **Autosubmit with Masking:** When Autosubmit with masking is enabled for a form, the form data is submitted automatically to the Web server, but the data sent to the Web browser over the network is masked for additional security.
- **Submitting with the help of advanced options:** If your form requires the execution of JavaScript when the form is submitted, you cannot use the Autosubmit options.This also means that single sign-on is disabled.

  To create a policy that allows autosubmitting for this type of form, you need to create the policy as described in "Creating a Form Fill Policy for Autosubmission" on page 738 and create two advanced options as described in "Configuring the Advanced Options for Autosubmission" on page 739.

Creating a Form Fill Policy for Autosubmission

1. In the Administration Console, click **Policies > Policies**.
2. Select the policy container, then click **New**.
3. Specify a display name for the policy and select **Access Gateway: Form Fill** for its type.
4. (Optional) Specify a description for the Form Fill policy.
5. In the **Actions** section, click **New**, then select **Form Fill**.
6. In the **Form Selection** section, select **Form Name** and specify **topaz** in the text box.
7. In the **Fill Options** section, specify all the input fields and select the options that you want.
8. In the **Submit Options** section, select **Auto Submit**.
9. Select **Enable JavaScript Handling**.
10. Select **Functions to Keep**, then specify the JavaScript functions that need to be retained when the form is being automatically submitted. For the example form, specify the following functions:

    ```
    function dvdRegisterSelect()
    function enableAll()
    function verify(f, bSubmitToSelf)
    function printThisView()
    function tpzDrillTable(a,b,c,d)()
    ```

11. Click **OK**.
12. Select **Statements to Execute** and specify the form action that needs to be performed when the form is submitted. For the sample form, specify the following statement:

    ```
    function executeJavaScript()
    {
        tpzDrillTable('','Login','0','listdetail');
    }
    executeJavaScript();
    ```

    You must perform this step in order to execute the functions configured in the **Functions to keep section** because the Access Gateway does not process HTML to include the `LAGSubmitForm()` function.
13. Click **OK**.
14. On the Policies page, click **Apply Changes**.

Configuring the Advanced Options for Autosubmission

When HTML forms contain complex JavaScript or VBScript, you must enable the following two advanced options:

- **#NAGGlobalOptions InPlaceSilent=on:** This enables single sign-on to certain Web sites that require the login page to remain as is without any modifications to its structure. This option is equivalent to `.enableInPlaceSilentFill` in the 3.1 SP4 Access Gateway Appliance.
- **#NAGGlobalOptions InPlaceSilentPolicyDoesSubmit=on:** This option enables form fill in HTML pages with complex JavaScript or VBScripts. This option is equivalent to `.enableInPlaceSilentFillNew` in the 3.1 SP4 Access Gateway Appliance.

To enable these options:

1. In the **Administration Console**, click **Access Gateways** > **Edit** > **Advanced Options**.
2. Add the following in the **Advanced Options** list:

   ```
   NAGGlobalOptions InPlaceSilent=on

   NAGGlobalOptions InPlaceSilentPolicyDoesSubmit=on
   ```

3. Click **OK**.

External Attribute Source Policies

Access Manager Appliance can be used as an identity provider for several different third-party service providers. Some of these service providers require attributes that are not part of the user store where the user is authenticated. The External Attribute Source policy allows you to retrieve attributes from the external sources.

You can configure this policy with rules to retrieve the attributes. A rule can contain all the conditions available for a policy. To enable this policy, you need to write a data extension class or provide a string constant value. The data extension class can be configured with constant and dynamic properties like any other policies.

For information about the structure and template of a data extension class and example code, see the Policy Extension API in the Access Manager 4.2 Developer Kit at NetIQ Access Manager 4.2 Developer Guide.

Enabling External Attributes Policy

An External Attributes policy must be enabled and configured before users can use the policy for fetching the attributes from external sources.

For assigning a policy to users, you must enable it for the Identity Server configuration.

1. In Administration Console, click **Devices > Identity Servers > Servers > Edit > External Attributes**.
2. Select the check box against the policy name and click **Enable**.
3. To disable the policy, select the check box against the policy name and click **Disable**.
4. To create a new policy, click **Manage Policies**.
5. After enabling or disabling policies, update the Identity Server configuration on the **Servers** tab.

Creating an External Attribute Source Policy

1. In the Administration Console, click **Policies > Policies**.
2. Click **New.**
3. Specify a name for the policy, select **Identity Server: External Attribute Source** for the type of policy, then click **OK**.
4. Specify the following details:

   Description: (Optional) Specify the purpose of this policy.

   Priority: Specify the sequence in which a rule is applied in the policy, when the policy has multiple rules. The highest priority is 1 and the lowest priority is 10.
5. In the **Actions** section, click **New**, then select **Fetch Attributes**.
6. Specify the following details:

   External Attribute Name: Specify the name of the attribute to be obtained through this policy.

   Value: Specify either **String Constant** or **Data Extension** for the attribute value.

   If you select **String Constant**, provide the value in the text box. The policy returns the string constant.

   If you select **Data Extension**, select the extension file from the list. The policy returns the attributes based on the logic defined in the class.

   For more information about policy extension, see “Adding Policy Extensions” on page 598.
7. Click **OK** twice, then click **Apply Changes**.
8. After creating an External Attribute Source policy, create a shared secret. This shared secret is used in configuring other policies or can be used by the Identity Servers in their attribute sets to retrieve attributes from external sources.

   For more information, see “Creating Shared Secret Names” on page 64.

External Attribute Source Policy Examples

You can use an External Attribute Source policy to retrieve attributes from external sources. You can create shared secrets from this policy. This shared secret then can be used in configuring other policies or can be used by the Identity Servers in their attribute sets to retrieve attributes from external sources.

An External Attribute Source policy must be enabled and configured before using the policy for retrieving the attributes from external sources.

For more information about how to create an External Attribute Source policy, see “Creating an External Attribute Source Policy” on page 740.

This section describe the usages of the External Attribute Source policy with the help of the following scenarios:

- “Scenario 1” on page 741
- “Scenario 2” on page 743

For information about sample codes for these examples, see Access Manager SDK Sample Code.

Scenario 1

e_Health is a Web portal for doctors. e_Health uses Med_Association as an external identity provider to verify whether the user is a doctor and obtain the user's professional code and specialization. Med_Association retrieves these details with the help of the NetIQ Identity Server.

Med_Association completes the following steps:

1. Write an External Attribute data extension class and use the required attribute to retrieve the professional code and specialization of user.

   For more information about data extension class, see “Adding Policy Extensions” on page 598.

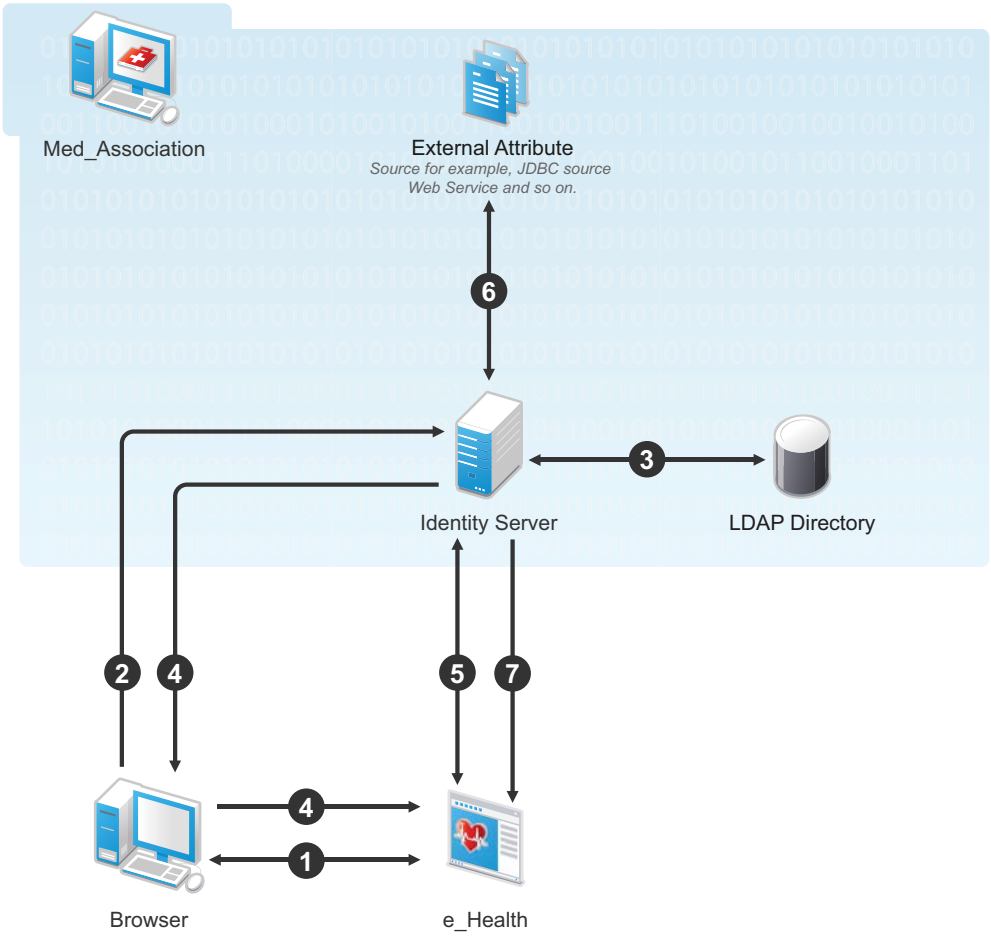
   For more information about data extension example code, see The Policy Extension API in the *NetIQ Access Manager 4.2 Developer Guide*.
2. Create an External Attribute Source policy for the data extension.

   For more information about how to import the data extension class and configure the External Attribute Source policy in the Identity Server, see Chapter , “External Attribute Source Policies,” on page 739.
3. Define a shared secret for the professional code and specialization.

   For more information, see “Adding Custom Attributes” on page 64.
4. Configure this shared secret for a service provider to be sent with authentication.

   For more information, see “Configuring the Attributes Sent with Authentication” on page 156.
5. The retrieved details that are professional code and specialization are sent to e_Health.

The following diagram illustrates this scenario:



Workflow:

1. User requests for access to e-Health through browser.
2. e_Health redirects the user's browser to the NetIQ Identity Server at Med_Association for authentication.
3. User logs in with providing credentials. User is authenticated with LDAP.
4. On the successful authentication, the Identity Server sends the assertion to e_Health.
5. e_Health verifies the assertion with Med_Association by using the back channel communication.
6. After verification, the NetIQ Identity Server retrieves the attributes (professional code and specialization) from external sources (for example, database) by using the External Attribute Source policy.
7. The Identity Server returns the response containing professional code and specialization in a shared secret attribute. If the user is not a doctor, external source returns null values in the shared secret attribute in the response.

   e_Health grants access to the user if it receives valid values for the attributes in the authentication response else it denies the access.

Scenario 2

Company XYZ is a customer of NetIQ Access Manger. The employees of this company get authenticated to the Identity Server. Each employee's mail attribute is retrieved from the user store. XYZ wants only user name part of the email address to be displayed on the Home page after authentication. This can be achieved by using the External Attribute Source policy.

XYZ completes the following steps:

1. Write an External Attribute data extension class and use the mail attribute as the parameter to the class.

   For more information about data extension class, see “Adding Policy Extensions” on page 598.

2. In the data extension class, read the email address and parse the name identifier in it and return as an attribute. For more information about data extension example code and example code for this scenario, see The Policy Extension API in the *NetIQ Access Manager 4.2 Developer Guide* guide.

3. Define a shared secret for the name field of the email address.

   For more information, see “Adding Custom Attributes” on page 64.

4. Create an External Attribute Source policy for the data extension.

   For more information about how to import the data extension class and configure the External Attribute Source policy in the Identity Server, see “Creating an External Attribute Source Policy” on page 740.

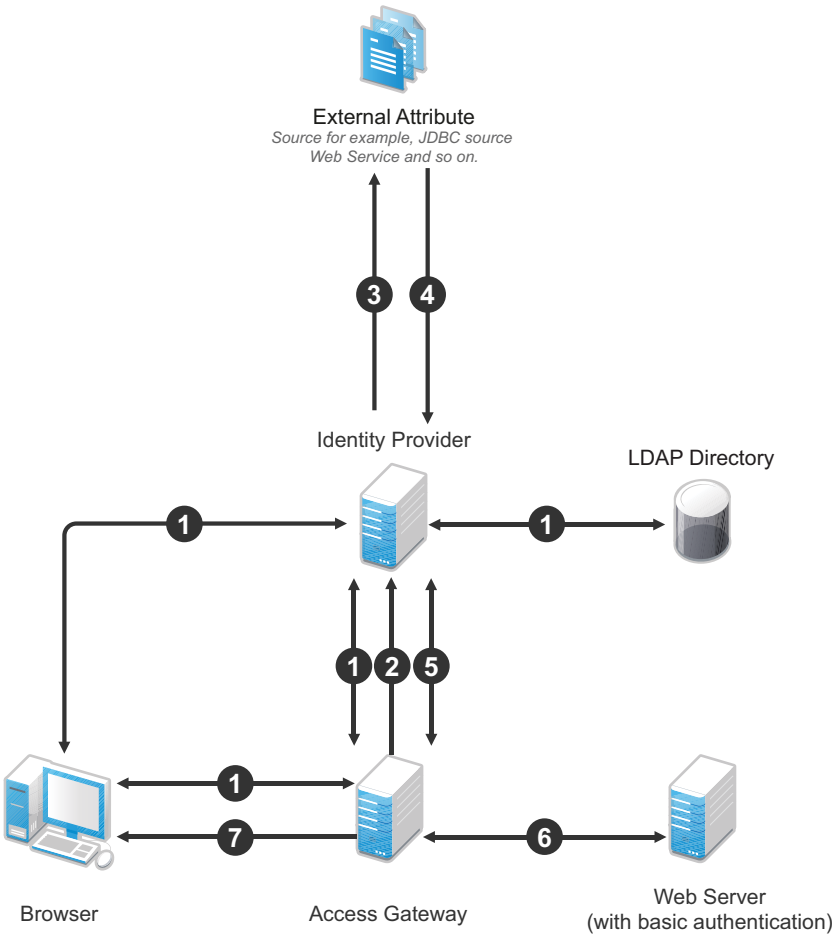
5. Create an Identity Injection policy.

   For more information, see Chapter , “Identity Injection Policies,” on page 691 and “Configuring a Custom Header Policy” on page 698.

6. The Identity Server sends the user ID part of email address to the Access Gateway.

   In turn, the Access Gateway or service provider sends this attribute to the configured Web server. For example, John is an employee of XYZ. He provides his email address, john@mail-domain.com, as his user name. After authentication, only John will be displayed on the Home page.

The following diagram illustrates this scenario:



Workflow:

1. User (through the browser) is requesting for a resource. The Access Gateway determines whether it is a protected resource and redirects the request to the Identity Server for authentication. The Identity Server authenticates with the LDAP servers and provides the assertion details to the Access Gateway. In turn, the Access Gateway verifies the assertion details.
2. The Home page in the resource is configured to display the user ID that has to be retrieved from the Identity Server.
3. The Identity Server determines whether the attributes can be retrieved from the external source. The Identity Server will send the required details to the external source (in this example, an email address).
4. The external source returns the data. In this example, user ID part of the email address.
5. The Identity Server sends the data that it has obtained from the external source to the Access Gateway.
6. The Access Gateway sends the data to the Web server.
7. The Web server returns the resource.

Risk-Based Policies

Topics include:

- “Configuring Risk-Based Authentication” on page 745
- “Configuring an Authorization Policy to Protect a Resource” on page 756
- “Enabling Auditing for Risk-Based Authentication Events” on page 756
- “Enabling Logging for Risk-Based Authentication” on page 756
- “Risk-Based Authentication: Sample Configuration” on page 757

For more information about risk-based authentication, see “Risk-based Authentication” on page 341.

Configuring Risk-Based Authentication

Configuring risk-based authentication involves the following steps:

1. Create a risk policy. For more information, see “Configuring a Risk Policy” on page 745.
2. Create a method for the risk-based authentication class. For more information, see “Configuring a Method for an Authentication Class” on page 749.
3. Create a contract for the risk-based authentication class. For more information, see “Configuring a Contract for an Authentication Class” on page 749.

You must consider the following points while configuring the risk-based authentication:

- A rule must be included in a risk policy. A rule can exist in multiple risk policies.
- A risk-based authentication class maps to only one risk policy and vice versa.
- If a rule condition is not met, the score associated with that rule is added to the risk score. If the rule condition is met, the specified action is executed.
- The risk level is determined based on the total risk score, that is the sum of the scores of all rule conditions that are not met.
- If a rule is configured to allow or deny access and exit the policy when a condition is met, the risk score is zero as other rules in the group are not evaluated.

Configuring a Risk Policy

Before creating a risk-based policy, determine the following to understand the criteria for defining a rule:

- ❒ The application or resource you want to protect.
- ❒ The parameters you want to assess during a login attempt.
- ❒ The risk score for each parameter.
- ❒ The risk levels for risk scores.
- ❒ The action for the risk levels.
- ❒ If you want to record the details of risk assessment.
- ❒ If you want to store history details from the risk assessment in MySQL, MicroSoft SQL Server, or Oracle database.

- ❒ If you want to perform profiling on user login events based on the geolocation of the user.
- ❒ If you want to assess the risk before a user attempts to login.

Configuring a risk policy includes the following three parts:

1. Adding a Risk Policy
2. Configuring Policy Rules
3. Configuring Risk Levels

Adding a Risk Policy

1 Click **Policies** > **Risk-based Policies** > **Risk Policy**.

If there is no risk policy configured, clicking **Risk Configuration** opens the Overview page.

2 Click the **Create Risk Policy** icon.

NOTE: To create an identical copy of any existing risk policy:

1. Click **Policies** > **Risk-based Policies** > **Risk Policy**.
2. Locate the policy you want to clone and click the **Clone Risk Policy** icon. All rules and risk levels configured for the existing policy are copied to the new policy.
3. Specify a name for the new policy and assign a cluster and an authentication class.

---

3 Under **Add Risk Policy**, specify the following details:

Risk Policy Name: Specify a name for the policy.

Policy Description: Describe the purpose of this policy.

Assign Policy To: Select the Identity Server cluster and then select an authentication class. You can select the class from the list of existing classes or you can create a new class.

NOTE: If you select an existing class, settings of the selected class are overwritten with values of this policy.

For creating a new class, perform the following steps:

Configuring a Risk-Based Authentication Class

3a Select either of the following options:

Create Risk-based Auth Class:

Calculates the risk score after authentication. For more information, see “Risk Assessment and Risk Mitigation after Authenticating a Login Attempt” on page 344.

Create Risk-based Pre-Auth Class

Calculates the risk score before authentication. For more information, see “Risk Assessment and Risk Mitigation before Authenticating a Login Attempt” on page 343.

NOTE: To modify an existing authentication class go to **Identity Server** > ***cluster*** > **Edit** > **Local** > **Classes**.

3b Specify a name for the class.

3c Select **Record User History** to record the user’s login details.

Before enabling this option, ensure that you have enabled recording user history in **Policies > Risk Configuration > User History** and configured a database. For more information, see “History:” on page 348 and “Configuring an External Database to Store User History” on page 360.

NOTE: The **Record User History** option is available only for `Risk-based Auth Class`.

3d Select **Use Cumulative Risk Score** to add current risk score of the session to this evaluation.

If you select this option, ensure that you have defined appropriate risk levels in this class to accommodate the cumulative value. For more information, see "Cumulative Scoring:" on page 349.

3e To send the user name, risk score, and risk level of a specific login attempt to an external REST interface, click **Score Sharing URLs** and specify the URL of the interface. The external REST interface uses this score information to perform additional actions on the user's identity.

(Optional) Specify the REST endpoint authentication credential. Whenever the risk score is sent to the REST endpoint, the endpoint sends these credentials as a basic authentication header. If the REST endpoint is protected by using basic authentication, then this credential is used.

If **Reduce Score** is enabled, the reduced risk score is sent to the REST endpoint for a successful additional authentication.

After enabling **Score Sharing URLs**, you must enable the identified risk scores for sharing.

You can enable two **Score Sharing URLs**. If the REST endpoint is down, a warning message is logged in the log file (Linux: `catalina.out`; Windows: `stdout.log`). If the REST endpoint is down during risk score sharing, the risk is not cached and will not be shared later.

NOTE: The **Score Sharing URLs** option is available only for `Risk-based Auth Class`.

3f Click **Save**.

4 Continue with Configuring Policy Rules.

Configuring Policy Rules

You can select a rule from the existing list or create a new rule and then assign it to the risk policy. You can assign multiple rules to a policy. The rules are executed in the top to bottom sequence. You can drag and drop to change the priority and sequence of rules. Rules for which action is defined as `Allow Access`, `Deny Access`, or `Exit with Risk Level as specified risk level` are executed as specified in the rule irrespective of the risk score accumulated for other rules that previously failed.

For each rule you want to add to a policy, you must perform the following steps:

1 To create a new rule, perform the following actions:

1a Click **Actions** > **Create Rule**.

1b Specify a rule name.

1c Select a rule definition. For details, see Step 3 in "Configuring Rules" on page 750.

1d Click **OK**.

1e Define actions for the rule.

Condition	Action
If rule condition is met	Select any of the following options: ◆ Proceed to Next Rule: The next rule in the execution sequence is executed. ◆ Allow Access and Exit Policy: No other rules of this policy is executed and the user will get access to the resource. ◆ Deny Access and Exit Policy: No other rules of this policy is executed and the user will be denied access. ◆ Exit with Risk Level as: Select a risk level. You can also create a risk level and then assign it to the rule. For information about creating a new risk level, see Configuring Risk Levels. No other rules in this policy are executed. The action specified for that risk level is executed.
If rule condition is not met, add risk score	Specify the risk score that will be stored when the rule evaluation fails.

1f Click **Save**.

NOTE: You can also create a rule here **Policies** > **Risk-based Policies** > **Rules** > **New**. See "Configuring Rules" on page 750.

2 To select a rule from the existing list, perform the following actions:

2a Under **Policy Rules**, click **Actions** > **Add Existing Rule**.

2b In **Risk Rule**, select the rule you want to add from the list.

2c Define actions for the rule. For details, see Step 1e on page 747.

2d Click **Save**.

NOTE: To validate the rule configuration and view the result when a condition is met, click **Actions** > **Toggle Validate**. For more information, see "Understanding How to Use the Validate Tool to Emulate Total Risk Score and Risk Levels" on page 362.

3 Continue with Configuring Risk Levels.

Configuring Risk Levels

1 Under **Risk Levels**, click **Actions** > **Add Risk Level**.

2 Specify the following details:

- **Risk Level**: Select a risk level to associate with the risk score. If you select **Other**, specify a name to identify the custom risk level.
- **Risk Score**: Specify a risk score to be associated with the risk level. The risk score indicates a value that is stored in the database after rule evaluation fails.
- **Action**: Select an action for this risk score.

  If you select **Additional Authentication** under **Action**, you can either select a class or a method to configure the step-up authentication. Typically, the step up to a method is used when branding, overwriting of users, or a change of userstore is required. If the userstore of the Additional Authentication is same as the risk-based authentication class and no additional branding is needed, then use a class.

- **Reduce Score:** After a successful additional authentication, you can configure to reduce the associated risk score. Specify the value that you want to reduce from the risk score in this field. For more information, see "Risk Score Reduction after a Successful Additional Authentication:" on page 349.
- **Share Score**: Select this option if you want to send the risk score of this risk level to the URL specified in **Score Sharing URLs** in the associated authentication class. You can share risk scores only for the risk levels configured for `Risk-based Auth Class`.

  This option is available only if at least one **Share Score URLs** is configured for the authentication class.

3 Click **Save**.

4 Continue with "Configuring a Method for an Authentication Class" on page 749.

Configuring a Method for an Authentication Class

Perform the following steps:

1 Click **Local** > **Method** > **New** to create a new method for the risk based authentication class.

2 Specify a name to identify the method.

3 Select the risk-based authentication class from **Class**.

4 Deselect **Identifies User**.

NOTE: If the policy is using `Risk-Based Pre-Auth Class`, this options must be selected at least in one method of the contract.

---

5 Select a user store from the list of **Available User Stores**.

6 Click **Finish**.

IMPORTANT: In a risk-based authentication class, properties configured for the risk-based authentication method are ignored. So, if you want to configure additional properties, add the property to the risk-based authentication class.

---

7 Continue with "Configuring a Contract for an Authentication Class" on page 749.

Configuring a Contract for an Authentication Class

Perform the following steps:

1 Click **Local > Contract > New** to create a new contract for the risk based authentication class.

2 Specify a name to identify the contract.

3 You can either use an existing authentication contract or create a new authentication contract. For example, for `Risk-based Auth Class`, you can add the default Name/Password – Form method as the first method and risk-based authentication method as the second method. For `Risk-based Pre-Auth Class`, the risk-based authentication method must be configured as the first method.

4 Click **Next** to configure a card for the contract. For more information about configuring contracts, see "Configuring Authentication Contracts" on page 288.

NOTE: If you want to use this contact in federation, ensure that you assign this contract to a protected resource.

Configuring Rules

Perform the following steps:

1 Click **Policies > Risk-based Policies > Rules > New**.

2 Specify a name for the rule.

3 Select a rule type in **Rule Definition** based on your requirement and specify the following details:

Rule Type	Procedure

IP Address	1. To manually add IP addresses, select **Manually enter the Datasource**. You can specify a single IP address, IP address range, IP address subnet, or upload a text file containing IP addresses. To specify the IPv4 subnets, use the Classless Inter-Domain Routing (CIDR) notation.
	Click **Add to List**.
	Sample text format
	`# Example IP List`
	`10.0.0.0`
	`172.16.0.0`
	`192.168.0.0`
	Each entry in the text files must be on a separate line.
	2. To consider the list of IP addresses provided by an external provider or an internal web service, select **Dynamically consume from the Datasource**.
	a. Specify **URL** of the provider.
	b. In **Connection Timeout**, specify the time in second. After this specified time, an unresponsive connection is closed. For example, 5 seconds.
	c. In **Refresh Interval**, specify the time in second. The connection will be refreshed at the specified interval. For example once in 86400 seconds.
	The external provider provides the list of IP addresses in text or JSON formats.
	Sample text format
	`# Example IP List`
	`10.0.0.0`
	`172.16.0.0`
	`192.168.0.0`
	Sample JSON format
	`["10.0.0.0","172.16.0.0","192.168.0.0"]`
	3. Specify how the conditions for the rule must match. The available options are **Is** and **Is Not**. For more information about Is and Is Not conditions, see Table 5-1, “Risk-based Authentication Terms,” on page 350.
	4. To validate the user history recorded in the database, select **Check user history**. You can use this option only when **Record user history** is enabled in the **User History** tab.
	IMPORTANT: You cannot specify the IP subnet in the IPv6 format. Instead, you can use the IP range condition and define it in the IPv6 format.

Rule Type	Procedure
Cookie	1. Specify the name of the cookie. 2. Specify the value of the cookie. The different search criteria that you can use are **Is** and **Is Not**. For more information about Is and Is Not condition, see Table 5-1, “Risk-based Authentication Terms,” on page 350. 3. [Optional] If the cookie is not found, but you want to create a cookie after the user authenticates, select **Create cookie if the user authenticates successfully**. a. Specify the validity of the cookie in days. b. Specify the path for the cookie. **IMPORTANT:** A cookie is set only when the user is authenticated by using second-factor authentication. The cookie is not created if the risk is assessed to be low and the user authenticates by using primary authentication method.
HTTP Header	1. Specify the HTTP header Name. 2. Specify the HTTP header value that you want to search for an HTTP header that includes a specific value. For example, if you want to search for an HTTP header that includes the value NetIQ, you can use the search criterion **Equals**. Whereas, if you want to query for an HTTP header that does not include the value NetIQ, use **Does Not Contain**.
User Profile	1. Select an LDAP attribute from the list. If you want to define a custom LDAP attribute, click **New**. 2. Specify how the conditions for the rule must be matched. 3. Specify the value of the attribute to be searched. For example, if you selected the LDAP attribute `birthDate` for rule creation, specify the birth date to be searched.
User Last Login	1. Specify the name of the last login cookie. 2. Specify the path for the cookie. 3. Specify the validity of the cookie in days. 4. If you want the cookie to be secured by HTTPS, enable **Secure Cookie**. 5. Specify the number of days the cookie can be accessed from the same device or system. This value must be less than the value in **Max Age**. 6. Specify the crypto key to encrypt the cookie. **IMPORTANT:** A User Last Login cookie is set only when the user is authenticated by using second-factor authentication. A User Last Login cookie is not created if the risk is assessed to be low and the user authenticates by using primary authentication method.
User Time of Login	1. Select Is/Is not condition based on your requirements. This determines how the conditions for the rule must be matched. 2. Specify the date and time of the user login. 3. To validate the user history recorded in the database, select **Check user history**. To use this option, enable **Record User History** in the **User History** tab.

Rule Type	Procedure
Device ID	1. Specify a name to identify the cookie. 2. Specify a path where the cookie has to be stored. 3. Specify the validity of the cookie in days. 4. If you want the cookie to be secure, select **Secure Cookie**. This ensures that the cookie is protected by HTTPS. 5. Specify the value that the cookie must contain. Select a value(s) from the list of cookie parameters. **NOTE:** For pre-authentication risk-based class, do not select username. The different search criteria you can use are **Is** and **Is Not**. For more information on how **Is** and **Is Not** condition can alter the search criteria, see Table 5-1, “Risk-based Authentication Terms,” on page 350 **IMPORTANT:** A Device ID cookie is set only when the user is authenticated by using second-factor authentication. A Device ID cookie is not created if the risk is low and the user authenticates by using the primary authentication method.
Geolocation	1. Specify the geolocation details. 2. Select Is/Is not condition based on your requirements. This determines how the conditions for the rule are matched. 3. To validate the user history recorded in the database, select **Check user history**. To use this option, select **Record User History** in the **User History** tab.

Rule Type	Procedure

External Parameter Configuration	1. Select **Negate Result** to reverse the result of the rule evaluation. For example, if you have defined this rule to get authentication details of a request using a specific IP address, you can use **Negate Result** to define the rule to not consider inputs originated from that IP address.
	2. Under External Parameter, add the following details for a parameter set:
	a. Name of the parameter.
	b. Specify the regular expression if required.
	For example, suppose an external source sends the following value for the `Virtual IPv4` parameter:
	`The Virtual IP address is 127.0.0.1`
	To extract the IP address from this string, you can specify a regular expression as follows:
	`(?:(?:25[0-5]|2[0-4][0-9]|[01]?[0-9][0-9]?)\\.){3}(?:25[0-5]|2[0-4][0-9]|[01]?[0-9][0-9]?)`
	This regular expression extracts the IP address `127.0.0.1` from the string and uses it for evaluating the configured condition. For more information about regular expression syntax, see the Javadoc for java.util.regex.Pattern.
	c. Select a relational or string operator to define a relationship between the parameter and parameter value. For example, whether a parameter must contain the specified parameter value or it must not be equal to the specified value. The available operators include **contains**, **does not contain**, **is equal to**, **is equal to ignore case**, **is greater than**, **is greater than or equal to**, **is less than**, **is less than or equal to**, **in between**, **is not equal to**, **is not equal to ignore case**.
	d. Specify a parameter value for evaluation.
	e. Click **Add Parameter** to add more parameters in this parameter set. You can define multiple parameters in a parameter set.
	3. (Conditional) Click **Add Parameter Set** to define additional parameter set. You can define multiple parameter sets.
	4. If you have defined two or more parameter sets, specify how the conditions for the parameters must match. The available options are **Or** and **And**. For more information, see Combination Rule in Table 5-1, “Risk-based Authentication Terms,” on page 350.
	For an example about configuring external parameter, see “Using External Parameters in Risk Assessment:” on page 349.
	NOTE: To use this rule, you must create a custom class to retrieve details from an external provider. For more information, see “User Information Methods” and “Creating an Authentication Class” in the *NetIQ Access Manager 4.2 Developer Guide*.

Rule Type	Procedure
Custom Rule	1. Specify a fully qualified name of the custom class for which you want to create a custom rule. For example: com.Company.test.MyCustomclass. 2. Select **Check user history** to check the user history details if the rule execution fails. 3. Select **Negate Result** if you want to reverse the results of rule execution. For example: if you have defined a rule to track authentication attempts from a specific geolocation, you can use **Negate Result** to define a rule to allow authentication if the user logging in is not from that geolocation. 4. Click **Add Property** to add custom properties and values.

4 Click **Finish**.

Configuring User History

Recording the user history involves the following steps:

1. Configuring an external database to log history records. For more information, see "Configuring an External Database to Store User History" on page 360.
2. Enabling user history. For more information, see "Enabling User History" on page 754.
3. Enabling user history while configuring a rule. For more information, see "Configuring Rules" on page 750.
4. Configuring user history for a risk policy that is linked to an authentication class. For more information, see "Configuring a Risk-Based Authentication Class" on page 746.

   When you choose to record user history details for a risk policy that is linked to an authentication class, you get the flexibility to segregate the history details as per your requirement.

   Consider a situation where you have configured the following two risk policies:

   - One risk policy assesses authentication requests from internal users in an organization.
   - Another risk policy assesses authentication requests from users external to the organization.

   To record the history details for internal users only, enable the recording of user history at the risk-based authentication class that is used to authenticate the internal users.

Enabling User History

1 Click **Policies** > **Risk-based Policies** > **User History**.

2 Select **Enable User History** to save the user session details in the database.

3 Under History Settings, you can either specify the number of days of history to consider during the rule execution or select the option to consider all historical records. For example, if you specify 10, it indicates that the details of last 10 days must be considered during the rule execution. If you do not specify the number of days, all historical records are considered for the execution.

NOTE: In Access Manager 4.1, you are required to specify the number of entries instead of the number of days. If you have configured the number of entries in Access Manager 4.1 and then upgrade to version 4.2, the configuration is ignored.

4 (Conditional) To store details in eDirectory, select **Built-in Data Store**.

NOTE: In a production environment, it is strongly recommended that you do not use eDirectory as the data store.

5 (Conditional) If you choose to save the session details in an external database, select **External Database**.

5a Specify the name to identify the driver.

5b Select the **Database Driver**. The driver path and dialect are displayed. You can change the driver and dialect details if required.

5c Specify the **Username** and **Password** to access the database.

5d Specify the **URL** to access the database.

NOTE: To configure MySQL as the database, ensure that the database URL is specified as `mysql://db_user:db_user@localhost/netiq_risk?autoReconnect=true`.

For information about configuring an external databases, see "Configuring an External Database to Store User History" on page 360.

6 Click **OK**.

Configuring Geolocation Profiling

1 Click **Policies** > **Risk-based Policies** > **Geolocation**.

2 Select **Enable Location Profiling** to fetch location data from a geolocation database. This helps to identify the location of the user based on the IP address details.

3 Select a **Geolocation Provider**. The available options are:

Database	Details
Neustar Service	◆ Specify the API Key and API Secret. ◆ Specify the Web Service URL.
Custom Provider	◆ Specify a name to identify the provider. ◆ Specify the fully qualified name of the JAVA class. ◆ Click **Add Property** to add properties to the custom class.

4 Click **OK**.

Configuring NAT Settings

You can configure the Identity Server to retrieve IP addresses in a NAT environment.

1 Click **Policies** > **Risk-based Policies** > **NAT Settings**.

2 Specify the name of the field to use for fetching the IP address of the client.

3 Specify the regular expression to retrieve the client IP address from the HTTP header value.

When you use the regular expression .* , the rule execution fails even if the client IP address exists in the list of multiple IP addresses. So, if you want to retrieve an IP address from a list of multiple IP addresses, modify the regular expression accordingly.

For example: If you specify the regular expression as .**?(?=,)**, the Identity Server considers the first IP address in the list to calculate risk.So, if the list of IP addresses are similar to 10.20.20.1,10.30.30.1,10.40.40.1, the regular expression .**?(?=,)** returns IP address 10.20.20.1.

4 Click **OK** to save the configuration.

Configuring an Authorization Policy to Protect a Resource

In an authorization policy, you can define a condition group that uses risk scores from the Identity Server to protect a resource.

Defining a Condition Group and Assigning Actions:

1 Select **Policies > Policies**.

2 Select the policy container and click **New**.

3 Specify a name for the policy and select **Access Gateway: Authorization** for the type of policy.

4 From the Condition Group, select **Risk Score.** For more information about Comparison, Value, and Result on Condition Error, see Risk Score.

5 Select an action. For more information about actions, see step 7 in Creating Access Gateway Authorization Policies.

6 Click **OK** to save the changes.

Enabling Auditing for Risk-Based Authentication Events

Access Manager logs the following risk-based authentication audit events:

- Risk-Based Authentication Succeeded
- Risk-Based Authentication Action Involved
- Risk-Based Authentication Failed

For information about how to configure Access Manager to send these events to a Novell Auditing Server, see “Enabling Identity Server Audit Events” on page 837.

Enabling Logging for Risk-Based Authentication

To enable logging for risk-based authentication, perform the following steps:

1 In the Administration Console, click **Devices** > **Identity Servers** > **Edit** > **Logging**.

2 Select **Enabled** under **File Logging**.

3 In the **Component File Logger Levels** section, specify any one of the following options for application logs:

- **Severe:** Logs serious failures that can stop system processing.
- **Warning:** Logs potential failures that have minimal impact on execution.
- **Info:** Logs informational events.
- **Verbose:** Logs static configuration information.

  The system logs any configuration errors under one of the three primary levels: Severe, Warning, and Info.
- **Debug:** Logs events for all other log levels: Severe, Warning, Info, and Verbose.

4 Click **OK**.

For more information, see “Identity Server Logging” on page 852.

Risk-Based Authentication: Sample Configuration

This section explains the use cases and their configuration steps demonstrated in the **Try Now** option on the **Risk Configuration** > **Overview** interface.

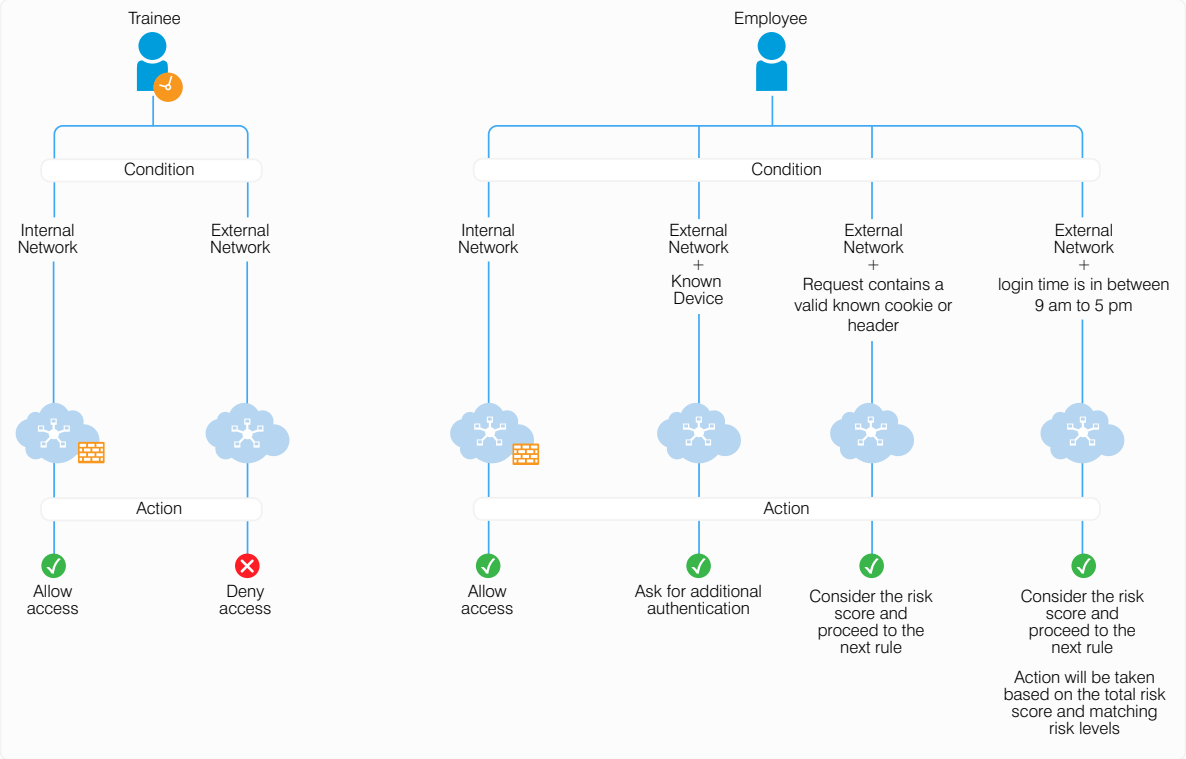
Let us assume a company named `Company1` wants to control access to its resources. There are two users. One is a permanent employee of `Company1` and another is a trainee at `Company1`. This configuration will refer the regular employee as `Employee` and the trainee as `Trainee`.

`Company1` wants to achieve the following actions:

- **Scenario 1:** `Trainee` or `Employee` is accessing the resource by using the internal network: Allow access.
- **Scenario 2:** `Trainee` is accessing the resource by using the external network: Deny access.
- **Scenario 3:** `Employee` is accessing the resource by using the external network with a known device: Ask for additional authentication.
- **Scenario 4:** Risk associated with accessing the resource from an external network can depend various conditions. The following are the conditions with equal weightage:
  - `Employee`’s request contains a cookie from the Intranet site or has a header from the Payroll site, indicating that `Employee` was successfully logged into these resources earlier.
  - Employee is logging in during normal work hours that is 9 am to 5 pm

  All conditions are evaluated. The risk decreases as the number of conditions met increases. The action performed depends on the risk score and associated risk level.

The following diagram illustrates the rule execution logic of the demo risk policy:



You can configure rules for these scenarios and arrange the rules in the order of priority on the UI. The rules are executed based on the priority from top to bottom. You can drag and drop rules on the UI to change their priority.

In this sample, the following five rules are created and executed in the same sequence:

Sequence of Execution	Rule Name	Action
1	DemoRule_InternalNetwork:	If `Trainee` or `Employee` is in the internal network, then allow access and exit the policy. If not, add risk score of 20 and proceed to the next rule.
2	DemoRule_TraineeUser	If `Trainee` is accessing from an external network, deny access and exit the policy. If `Employee` is accessing from an external network, add risk score of 20 and proceed to the next rule.
3	DemoRule_KnownDevice	If `Employee` is using a known device to access the resource from an external network, prompt for an additional authentication and exit the policy. If the `Employee` is using a unknown device to access the resource from an external network, add risk score of 20 and proceed to the next rule.
4	DemoRule_Combo	If `Employee` is accessing with a cookie from the payroll site or HTTP Header value contains loggedIn, proceed to the next rule with the total risk score accumulated due to the failure of the above three rules. For example, assume the risk score for each rule is 20. If the condition for this rule is met, then proceed to next rule with a risk score 60. If the condition for this rule is not met, then proceed to next rule with a risk score of 80. **NOTE:** To use this rule, you must set cookies or headers per domain with a path of /, so that the Identity Server can receive them.
5	DemoRule_TimeOfLogin	If `Employee` is logging in using an external network and time is in between 9 AM to 5 PM, proceed to the next rule. The risk score will depend on whether the condition of the DemoRule_Combo was met. If the conditions of both DemoRule_Combo and DemoRule_TimeOfLogin rules fail, the total risk score will be 100 and access will be denied.

The following steps have been performed to configure the demo risk policy for the identified scenarios:

1. Go to **Policies** > **Risk-based Policies** > **Risk Policy**.
2. Click the **Create Risk Policy** icon.
3. Under Add Risk Policy, specify the following details:

   Risk Policy Name: Specify `Demo_RiskPolicy`.

   Policy Description: Specify the purpose of this policy.

Assign Policy To: Select the Identity Server cluster and then configure an authentication class.

3a Select **Create Risk-based Auth Class**.

3b Specify **Class Name** as `Demo_RBAAuthClass`.

3c Click **Save**.

4 Create the following rules:

Under **Policy Rules**, click **Create Rule** and specify the following values:

4a DemoRule_InternalNetwork

- **Rule Name**: `DemoRule_InternalNetwork`
- **Rule Definitions:** `IP Address Rule`
- Specify the IP address range as `121.1.1.1` - `121.121.255.255` and click **OK**
- **If rule condition is met, then**: `Allow Access and Exit Policy`
- **If rule condition is not met, add risk score:** `20`
- Click **OK**.

4b DemoRule_TraineeUser

- **Rule Name**: `DemoRule_TraineeUser`
- **Rule Definitions:** `User Profile`
- Select `EmployeeType`, `Equals`, and then specify `Trainee`. Click **OK**
- **If rule condition is met, then**: `Deny Access and Exit Policy`
- **If rule condition is not met, add risk score**: `20`
- Click **OK**.

4c DemoRule_KnownDevice

- **Rule Name**: `DemoRule_KnownDevice`
- **Rule Definitions:** `Device ID Rule`
- Do not change the default values and click **OK**
- **If rule condition is met, then**: `Exit with Risk Level as`
- **Select Risk Level**: `Medium`
- **If rule condition is not met, add risk score**: `20`
- Click **OK**

4d DemoRule_Combo

- **Rule Name**: `DemoRule_Combo`
- **Rule Definitions:**

  Rule 1: Cookie Rule

  Cookie Name: `IntranetCookie`

  Cookie Value: `is test 12`

  Rule 2: `HTTP Header Rule`

  HTTP Header Name: `PayrollAccessHeader`

  HTTP Header Value: `Contains loggedIn`
- **Combination Rule Definition**: In Condition Group, click **Assign Rules** and then select both rules. Select `OR` in **Group Operator**
- **If rule condition is met, then**: `Proceed to Next Rule`

- **If rule condition is not met, add risk score**: `20`
- Click **OK**

4e DemoRule_TimeOfLogin

- **Rule Name**: `DemoRule_TimeOfLogin`
- **Rule Definitions:** `User Time of Login Rule`

  User time of login: `is`

  Day: `Monday to Friday`

  Time: `9 AM to 5 PM`
- Click **OK**
- **If rule condition is met, then:** `Proceed to Next Rule`.
- **If rule condition is not met, add risk score**: `20`
- Click **OK**

5 Under Risk Levels, click **Actions** > **Add Risk Level** and create the following risk levels:

- **Low**

Field	Value
Risk Score	Less than 40
Risk Level	Low
Action	Allow Access

- **Medium**

Field	Value
Risk Score	Between 40 and 80
Risk Level	Medium
Action	Authenticate using Trust levels

- **High**

Field	Value
Risk Score	Greater than 80
Risk Level	High
Action	Deny Access

6 Click **OK**.

7 Create an authentication method. For more information, see “Configuring a Method for an Authentication Class” on page 749.

8 Create a contract. For more information, see “Configuring a Contract for an Authentication Class” on page 749.

9 Assign the contract to the protected resource.

7 Enabling Mobile and Web Access

enables you to provide web and mobile access to protected applications and resources through the MobileAccess feature. MobileAccess includes a configuration portion in the Administration console as well as the MobileAccess app for mobile devices. Configurable options in MobileAccess include the following:

- Which applications users must be able to access through appmarks
- Whether users can access an application through a desktop browser or a mobile device, or both
- What the preferred viewer for the application on the mobile device is
- Whether users are required to provide a PIN to use the MobileAccess app on their mobile device, and if so, whether they are required to re-enter the PIN after a period of inactivity

Appmarks are essentially bookmarks for Access Manager resources. After you have configured an Access Manager resource, you configure one or more appmarks to enable users to access the resource in different ways. You can create an appmark for:

- Bookmarks
- Mobile apps
- Service providers
- Protected resources

The MobileAccess app that users install on their mobile devices enables them to access Access Manager resources from those devices. Administrators can also make the MobileAccess app available to users in a private corporate store. After users have installed the app and registered their device, they can access the resources using the assigned authentication methods.

Administrators can deregister user mobile devices through the administration console on the User Devices page. So, if a registered mobile device is lost or stolen, or an employee leaves the company, you can ensure that unauthorized users cannot access corporate resources. Users can also deregister their own mobile devices if necessary, either from their device or from the user portal page.

User Requirements for MobileAccess

The following are the minimum requirements for users to use MobileAccess.

- **Mobile Devices:** The following are the supported mobile devices and versions for MobileAccess:
  - iPhone with iOS 8.*x* or 9.*x* or later
  - iPad or iPad mini with iOS 8.*x* or 9.*x* or later
  - Android phones and tablets with KitKat 4.4 or Lollipop 5.x
- **Browsers:** The following are the supported browsers for MobileAccess:
  - Mozilla Firefox (latest version)
  - Google Chrome (latest version)
  - Microsoft Internet Explorer 11 (no Edge support)
  - Apple Safari (latest version)

Configuring Appmarks

Appmarks function like bookmarks in browsers. As with bookmarks, if the resource for the appmark changes, you must change the appmark to match the modified source. For example, you have an appmark for a protected resource and the protected resource changes, you must edit your appmark and select the modified protected resource again to receive the changes.

You can configure one or more appmarks for a resource. However, that appmark is specific to a single IDP cluster. You can also configure appmarks for users to use with only a browser. If you want users to use the mobile apps, you must first configure MobileAccess. For more information, see “Configuring MobileAccess” on page 764.

When you configure an appmark, you specify whether you want the application to launch in a desktop browser or on a supported mobile device, or both. If you configure a single appmark to display in both a desktop browser and on a mobile device, the appmark will have the same name, but you can customize the icons so they are different. Appmarks offer significant flexibility, enabling you to customize your users’ experience using different view options and variables.

When you configure a new appmark to display on a mobile device, the user must do a refresh on the application page on the mobile device before the appmark appears.

To create an appmark:

1. In the Administration Console, click **Appmarks**.
2. (Conditional) If you have not already configured an Access Manager resource, create a resource, such as a service provider or a protected resource.
3. In the **Cluster** field, select the appropriate IDP cluster that hosts the protected resource.

   If you have only one IDP cluster, the Administration Console automatically selects that IDP cluster.
4. Click the plus sign (+) to create an appmark.
5. Create a new appmark using the information located in, “Understanding Appmark Options” on page 763.
6. Click **Save**.

Repeat this process for each appmark you want to create. In order for the mobile appmark to work, you must configure MobileAccess. For more information, see “Configuring MobileAccess” on page 764.

Creating Multiple Appmarks for an Application

Application connectors can have multiple appmarks. For example, you might create several appmarks for different Office 365 or Salesforce applications. You can create a new appmark, or you can duplicate an existing appmark to save time, especially if you want to create several appmarks and just change one or two options on each one.

To duplicate a new appmark:

1. In the Administration Console, click **Appmarks**.
2. Click the menu on the upper left corner on the appmark you want to copy, then click **Duplicate**.
3. Edit the **Name** field because it is pre-populated with `COPY_$(DisplayName)`. You have the following options:
   - You can accept this default name. (However, note that “COPY_” will be part of the name.)
   - You can change the display name by manually editing the text.

4 Specify whether the application must be accessible from a desktop browser or a mobile device, or both, and complete the appropriate fields. For more information about available options, see "Understanding Appmark Options" on page 763.

5 Click **Save**.

Understanding Appmark Options

You configure appmarks in the Administration Console under **Appmarks**. On each appmark that you create, there is a menu on the upper left corner that allows you to edit, duplicate, or delete the appmark. Clicking **Edit** or clicking the appmark allows you to edit the appmark. Clicking the plus sign (+) at the top of the page, allows you to create a new appmark. When you create or duplicate an appmark, the following options are available.

Table 7-1 *Appmark Options*

Option	Description
Name	The display name for the appmark. If you want different display names for the appmark on the desktop browser page and on mobile devices, you must create a copy of the appmark and change the name.
Description	The description appears as hover text that users see for the appmark on the user portal page.The platforms are Desktop, iOS, and Android.
Change Image	The image that Access Manager uses for the appmark for all platforms, unless you override this image with a another image when configuring the different platforms. The maximum image size is 200 x 200 pixels and the ideal image size is 100 x 100 pixels. You can use an image from the Image Gallery or upload your own image.
Roles	(Optional) The Access Manager role that the users must have in order to see the appmark. If you do not select a role, all users can see the appmark on the user portal page. If you add a role, only users with that role can see the appmark. If you add multiple roles, users in any of those roles can see the appmark. For example, if you add Sales and Managers, the users must be in Sales or Managers, not Sales and Managers, to see the appmark.
Type	The Access Manager resource type that the appmark represents. There are many different types of resources: ◆ **Bookmark:** The URL of a website or document. ◆ **Mobile App:** The URL that launches the application at the mobile device. When the user opens the MobileAccess app on the mobile device and taps the appmark, MobileAccess opens the mobile app itself. ◆ **Service Provider:** A trusted service provider such as Google or Salesforce that you configured before creating the appmark. If you have enabled the **Allow any target** in the **Intersite Transfer Service** for the service provider, you can specify a target override URL as well. ◆ **Protected Resource:** A protected resource on a trusted Access Gateway cluster that you have created prior to creating the appmark. The protect resource roles control access to the appmark, not the appmark roles. The appmark roles only provide visibility of the appmark to the users.
Enable	Select the user platforms where the appmark will be visible. The platforms are Desktop, iOS, and Android.

Option	Description
Desktop	Allows you to override the behaviors of the desktop appmark. For example, you can add a different icon for the desktop appmark so it sizes differently than the iOS appmark. You can use an image from the Image Gallery or upload your own image.The options to override are: ◆ Image ◆ URL
iOS and **Android**.	The options are the same whether you select iOS or Android. You can add a unique image or URL for the iOS and Android appmarks so that these appmarks appear differently from the desktop appmarks. The appmarks also have additional options not available for the user portal page. ◆ **Launch with:** Specifies how to launch the application on the mobile device. Options include the following: ◆ **Chrome:** When the user opens the MobileAccess app on the mobile device and taps the appmark, the MobileAccess app launches Chrome and directs it to the application. If Chrome is not installed on the mobile device, the user is taken to the App Store or Google Play to install it. ◆ **Internal viewer:** When the user opens the MobileAccess app on the mobile device and taps the appmark, the MobileAccess app opens an embedded HTML viewer and directs it to the application. This view is similar to the Safari and Chrome options, except that the user does not have to leave the MobileAccess window. The application opens within the MobileAccess app window, and the user can tap the app name (as defined by the administrator when configuring MobileAccess) on the navigation bar in the top left corner of the screen to go back to the app home page and easily switch to another protected resource. ◆ **Safari (iOS only):** When the user opens the MobileAccess app on a mobile device and taps the appmark, the MobileAccess app launches Safari and directs it to the application. ◆ **User Choice (Android only):** When the user opens the MobileAccess app on a mobile device and taps the appmark, the MobileAccess app allows the user to choose what browser launches. ◆ **App Installer URL:** (Optional) You can use this option if you selected the **Mobile App** type. This is the URL to install the application if it is not already installed on the mobile device.

Configuring MobileAccess

After you have created appmarks, you must configure MobileAccess to enable users to register their mobile devices through the MobileAccess app (iOS or Android). Users can access the appmarks through a browser on a desktop or a mobile device without enabling MobileAccess.

NOTE: MobileAccess communicates only over an HTTPS connection. MobileAccess does not work with HTTP.

To configure MobileAccess:

1. In the Administration Console, click **MobileAccess**.
2. Select the IDP cluster that contains the appmarks you want to enable for the MobileAccess app.
3. Select **Enable MobileAccess** to enable users to register their devices if they have the MobileAccess apps installed.
4. In the **Device display name** field, specify your company name. This name appears in the bar at the top of the MobileAccess app window on users' mobile devices.
5. In the **Roles** field, select the roles that users must have to be able to register their devices through the MobileAccess app.

   If you do not select a role, all users can register their devices through the MobileAccess app. If you add a role, only users with that role can register their devices through the MobileAccess app. If you add multiple roles, users in any of those roles can register their devices through the MobileAccess app. For example, if you add Sales and Managers, the users must be in Sales or Managers, not Sales and Managers, to register their devices through the MobileAccess app.
6. In the **Mobile device registration contract** field, select the contract that users will see to register their devices through the MobileAccess apps. Currently, MobileAccess only supports a user name and password contract.
7. In the **Methods satisfied by mobile authentication** field, select the authentication methods that are satisfied after users have successfully registered a mobile device.
8. In the **Password Prompt** field, select how long users can continue to use an authenticated password on mobile devices before re-authentication is required.
9. In the **PIN Prompt** field, select whether users must set a PIN for the MobileAccess app on their mobile devices, and whether they must re-enter the PIN after a period of inactivity. You can change this requirement at any time. For more information, see "Understanding the MobileAccess PIN" on page 767.
10. Click **Save**.
11. Repeat the procedure for each IDP cluster that contains appmarks.

Helping Users Register Their Mobile Devices

Users can install the NetIQ MobileAccess app on a mobile device to use MobileAccess with Access Manager. A user can register a device with multiple providers by setting up separate accounts for each one. If a user registers a device with multiple providers, the user must select the account to use for a session from the providers list on the device. By default, the app connects the user to the first provider in the list.

As an administrator you can provide two different ways for users to register a mobile device with Access Manager.

Registering iOS Devices

There are two different ways you can provide to users to register an iOS mobile device.

- **Manual:** Send your users an email with the URL to the IDP server including the correct port number. The users enter the IDP URL in the MobileAccess app by tapping the menu in the upper left corner of the MobileAccess app, then tap **Accounts** > **+** to enter the URL of the IDP in the **Providers** field. The URL is:

  `https://IDP_server_dns_name:port`

For example:

```
https://idp.acme.com:8443
```

- **Embedded Link in Email:** You can send your users the following link. When the users click the link, the link launches the MobileAccess app and the user is presented with the Sign In page. The user signs in using their corporate credentials to access the appmarks available in the MobileAccess app. The link is:

  ```
  comnetiqauth://x-callback-url/register?providerUrl=https://
  IDP_server_dns_name:port/
  ```

  For example:

  ```
  comnetiqauth://x-callback-url/register?providerUrl=https://idp.acme.com:8443/
  ```

  NOTE: Email client can prevent embedded links from working.

Registering Android Devices

There are two different ways you can provider to users to register an Android device.

- **Manual:** Send your users an email with the URL to the IDP server including the correct port number. The users enter the IDP URL in the MobileAccess app by tapping the menu in the upper right corner of the MobileAccess app, then tapping **Manage Accounts** > **+** to enter the URL of the IDP in the **Provider** field. The URL is:

  ```
  https://IDP_server_dns_name:port
  ```

  For example:

  ```
  https://idp.acme.com:8443
  ```

- **HTML Page with Anchor Link:** You can also create an HTML page that contains an anchor link that users click on to have the **Provider** field populated for them. The format of the anchor link is:

  ```
  <html>
    <body><a href="intent://x-callback-url/register?providerUrl= https://
  IDP_server_dns_name:port#Intent;scheme=comnetiqauth;package=com.netiq.mobileac
  cessforandroid;end;">Register</a></body>
  </html>
  ```

  For example:

  ```
  <html>
    <body><a href="intent://x-callback-url/register?providerUrl= https://
  idp.acme.com:8443#Intent;scheme=comnetiqauth;package=com.netiq.mobileaccessfor
  android;end;">Register</a></body>
  </html>
  ```

Installing MobileAccess on a Mobile Device

After you have created appmarks and enabled MobileAccess in the Administration Console, users must install the MobileAccess app on supported mobile devices before they can access any resources that have been configured for mobile access. For more information, see *NetIQ® Access Manager Mobile Users QuickStart*.

Understanding the MobileAccess PIN

Access Manager administrators can require users to set a PIN on their mobile devices as a security measure to prevent unauthorized users from accessing protected resources through the MobileAccess app. Administrators can also specify whether users must re-enter the PIN after a period of inactivity on the device.

Users must install the MobileAccess app on the mobile device before they can set the PIN. If a PIN is required, the MobileAccess app prompts users to set the PIN the first time they open the app. Otherwise, users can set, change, or remove the PIN any time by accessing the Settings page from the MobileAccess app.

NOTE: The MobileAccess PIN is unrelated to the built-in device passcode, which is designed to protect other resources on the mobile device.

Even if the administrator does not require users to set a PIN, users can optionally set a PIN on their device. The PIN can be different for each mobile device the user registers. The PIN is not stored anywhere other than on the device itself.

Administrators can change the **PIN Prompt** setting any time in the Administration Console. If the administrator specifies that a PIN is required after a mobile device has already been registered, the next time the user launches the MobileAccess app on the mobile device, MobileAccess prompts the user to set a PIN. The app then prompts the user for that PIN each subsequent time the user accesses the app. If the administrator initially requires users to set a PIN and then changes that requirement, users can remove the PIN from their device. However, MobileAccess does not notify users if a PIN is no longer required.

Whether the Access Manager administrator requires users to set a PIN or a user chooses to set a PIN, by default users can enter their PIN incorrectly five times. On the fifth attempt, the application deregisters the mobile device and removes the current PIN. The user must then reregister the device and reset the PIN. For more information, see “Deregistering Your Device” in the *NetIQ® Access Manager Mobile Users QuickStart*.

Managing Mobile Devices

Access Manager administrators can manage and deregister user mobile devices in the Administration Console. So, if a registered mobile device is lost or stolen, or an employee leaves the company, you can ensure that unauthorized users cannot access corporate resources.

Users can also deregister their own mobile devices, either from their device or from the user portal page after they log in. A mobile device that has previously been deregistered can be reregistered by the same user. However, for a different user to use the deregistered mobile device, the user must delete and reinstall the MobileAccess app on the device before reregistering the device.

Use the information in the following sections to help you manage mobile devices:

Deregistering Mobile Devices

If you are logged in to the Administration Console as an administrator, you have the option to search for and deregister devices that are registered to other users. Users can manage their own devices from the user portal page after they log in.

To deregister mobile devices:

1. In the Administration Console, click **User Devices**.
2. Select the IDP cluster that contains the user.
3. If you want to search for the devices belonging to a particular user, enter the user name in the **User** field.

   or

   Browse the list of devices the Administration Console displays.
4. Click the **Delete** icon next to the device you want to deregister, then click **OK** on the confirmation message.

After a mobile device has been deregistered, the device can be registered to a new user. However, the MobileAccess app on the device must first be deleted and reinstalled. For more information, see "Deleting and Reinstalling the MobileAccess App on a Device" on page 768.

Deregistering a Mobile Device as a User

Users who have previously registered a mobile device can deregister the device if necessary. For more information, see "Deregistering Your Device" in the *NetIQ® Access Manager Mobile Users QuickStart*.

NOTE: Users can uninstall the MobileAccess app on a mobile device after the device has been deregistered. However, if the MobileAccess app is uninstalled without the device first being deregistered, the device continues to appear on the Devices page. The administrator or user can delete the device from the Devices page in the Administration console.

Deleting and Reinstalling the MobileAccess App on a Device

After a mobile device has been deregistered, the MobileAccess app on the device must be deleted and reinstalled before a different user can reregister the device.

To delete and reinstall the MobileAccess app on a device:

1. Follow the instructions to uninstall the MobileAccess app:
   - iOS (http://www.apple.com/support/iphone/assistant/application/#section_5)
   - Android (https://support.google.com/googleplay/answer/2521768?hl=en)
2. Reinstall the MobileAccess app. For more information, see "Installing the MobileAccess App" in the *NetIQ® Access Manager Mobile Users QuickStart*.

Installing the Access Manager Certificate Authority Certificate on a Mobile Device

You can add the Access Manager certificate authority (CA) certificate to an iOS mobile device by performing the following steps:

NOTE: This procedure does not work in Chrome.

To install a self-signed certificate on the mobile device:

1. Take the Access Manager CA certificate and email it to the user who has an email account configured on the mobile device. Alternatively, you could put it on a web or FTP site that is accessible from the mobile device.
2. Open the email, website or FTP site on the mobile device and tap the certificate attachment.
3. In the Install Profile window, tap **Install**.
4. Read the warning, then tap **Install**.
5. Verify that the certificate reads **Trusted** with a green check mark in the Profile Installed window.
6. (Conditional) If the certificate is not trusted, something is wrong with the certificate and the MobileAccess app will not work. Go back and try to generate the certificate again.
7. Tap **Done**.
8. Specify the appliance DNS name in the Safari address bar and ensure that there is no warning about an untrusted certificate.

Changing the Branding of the User Portal Page

The Administration Console allows you to change the default branding of the user portal page so your users see only your company's information. To do complex customization, you can still edit the JSP file to make changes. For more information, see "Customizing The Identity Server" on page 192.

To make simple branding changes on the user portal page:

1. In the Administration Console, click **Branding**.
2. Select the appropriate IDP cluster.

   Just as appmarks are unique per IDP cluster, the branding is unique per IDP cluster.
3. Click **Change Image**, then either select an image from the image gallery or upload your own image.
4. In the text field, specify a name that is appropriate for your company.
5. Specify a color name, a RGB value, or a HEX number for the appropriate color for the banner. For example, enter #ffffff, Aquamarine, or rgb(26,155,130).

   For a full list of the supported color names, see HTML Color Names.

   For a solid background color, leave the **Right Background Color** field blank. You can also use a valid CSS background style syntax in either background definition field, but not both fields. For example, enter linear-gradient (to right, red, blue).

   IMPORTANT: To properly brand the MobileAccess applications, use only one HEX number for each color and no advanced CSS features.

   ---
6. Click **Save**.
7. Repeat the procedure for each IDP cluster.

To change the new user portal page beyond the branding changes you must perform manual steps and customization of `.jsp` files. For more information, see "Customizing The Identity Server" on page 192.

8 High Availability and Fault Tolerance

Topics include:

- “Installing Secondary Versions of Access Manager Appliance” on page 771
- “Configuration Tips for the L4 Switch” on page 774
- “Setting up L4 Switch for IPv6 Support” on page 779
- “Using a Software Load Balancer” on page 783

Installing Secondary Versions of Access Manager Appliance

The Administration Console contains an embedded version of eDirectory, which contains all configuration information of Access Manager Appliance. It also contains a server communications module, which is in constant communication with the Access Manager modules. If the Administration Console goes down and you have not installed any secondary consoles, your Access Manager components also go down and your protected resources become unavailable.

- “Prerequisites” on page 771
- “Understanding How Consoles Interact with Each Other and with Access Manager Devices” on page 773

Prerequisites

- ❒ An L4 server is installed. The LB algorithm can be anything (hash/sticky bit), defined at the Real server level.
- ❒ Persistence (sticky) sessions enabled on the L4 server. You usually define this at the virtual server level.

NOTE: If Access Manager Appliance is configured with public and private interface, the back channel communication will use the private interface. To allow this back channel communication on private interface, modify the NAM-RP configuration to listen on private and public interfaces. For more information, see “Managing Reverse Proxies and Authentication” on page 94.

Configuration Notes

A Note about Layer 4 Switch: A cluster of Access Manager Appliances must reside behind a Layer 4 (L4) switch. Clients access the virtual IP address of the cluster presented on the L4 switch, and the L4 switch alleviates server load by balancing traffic across the cluster.

Whenever a user accesses the virtual IP address assigned to the L4 switch, the system routes the user to one of the Access Manager Appliances in the cluster, as traffic necessitates.

IMPORTANT: You must not use a DNS round robin setup instead of an L4 switch for load balancing. The DNS solution works only as long as all members of the cluster are working and in a good state. If one of them goes down and traffic is still sent to that member, the entire cluster is compromised and all devices using the cluster start generating errors.

Services of the Real Server: A user's authentication remains on the real (authentication) server cluster member that originally handled the user's authentication. If this server malfunctions, all users whose authentication data resides on this cluster member must re-authenticate unless you have enabled session failover. For more information about this feature, see "Configuring Session Failover" on page 55.

Requests that require user authentication information are processed on this server. When the system identifies a server as not being the real server, the HTTP request is forwarded to the appropriate cluster member, which processes the request and returns it to the requesting server.

A Note about Service Configuration: If your L4 switch can perform both SSL and non-SSL health checks, you must configure the L4 switch only for the services that you are using in your Access Manager configuration. For example, if you configure the SSL service and the non-SSL service on the L4 and the base URL of your Identity Server configuration is using HTTP rather than HTTPS, the health check for the SSL service fails. The L4 switch then assumes that all the Identity Servers in the cluster are down. Therefore, ensure that you enable only the services that are also enabled on the Identity Server.

A Note about Alteon Switches When you configure an Alteon switch for clustering, direct communication between real servers must be enabled. If direct access mode is not enabled when one of the real servers tries to proxy another real server, the connection fails and times out.

To enable direct communication on the Alteon:

1. Go to **cfg > slb > adv > direct**.
2. Specify *e* to enable direct access mode.

Installing a Secondary Access Manager Appliance

1. Insert the CD containing the software.

   Most of the installation process is same for a secondary appliance as for a primary. If this is a second or third appliance, the Administration Console will be configured for the fault tolerance. While installing a secondary appliance:

   - Deselect the **Primary** check box.
   - Specify the IP address of the primary Administration Console.
   - Specify the user name and password of the primary Administration Console.

   Installation of the secondary appliance becomes interactive after the installation of operating system in the following cases:

   - (Conditional) if this is the fourth appliance: The number of Administration Consoles in a cluster is restricted to three. If more appliances are added into the cluster, the system will ask whether you want proceed with the installation of rest of the components other than Administration Console.
   - (Conditional) if time is not synchronized between the primary and secondary appliances. The system will prompt a message asking you to re-try the time synchronization or to proceed without synchronization.

   If you have firewalls separating your Identity Servers or your L4 switch does not support port translation, you can use iptables to translate the port

Configure the details on the Administration Console Configuration page as specified in step 9 in Installing Access Manager Appliance in the NetIQ Access Manager Appliance 4.2 Installation and Upgrade Guide.

2 Continue with the installation process.

The Identity Server and the Access Gateway from the secondary appliance are automatically clustered with the primary appliance. If this is second or third secondary appliance, the configuration store will be configured for the fault tolerance. Install at least one secondary console.

After successful installation, the appliance points to the Access Manager Appliance's IP address for the Web server, and the Identity Server points to the local user store. If a cluster is configured for Access Manager Appliance and if primary appliance is down, you cannot authenticate because the user store is on primary and they cannot access the resources because it points to the Web server on primary. Hence, it is advised to change the IP address of the Web server configured in the master proxy service to point to your test or production Web server, and change the Identity Server's configuration to point to an external user store.

Understanding How Consoles Interact with Each Other and with Access Manager Devices

Primary and secondary consoles use eDirectory synchronization to keep their configuration databases current.

WARNING: As long as the primary console is running, all configuration changes must be made at the primary console. If you make changes at both a primary console and a secondary console, browser caching can cause you to create an invalid configuration.

Access Manager Appliance devices use the secondary console only when the primary console is down. Therefore, if a secondary console goes down while the primary console is running, devices are notified. But they continue to run by using the primary console for configuration information. The secondary console can be down for as long as required to fix the problem without affecting other Access Manager Appliance devices.

When the primary console goes down, all of the devices discover this and switch to using the secondary console. This can take a few minutes, because each device has its own trigger for checking in with the Administration Console. After the device has switched to using the secondary console, it continues to run just as it did when it was communicating with the primary console. When the primary console comes back online, all devices discover this and switch back to using the primary console. Again, this can take a few minutes.

Not all tasks are available from the secondary console:

- "Tasks Requiring the Primary Console" on page 773
- "Tasks Available from the Secondary Console" on page 774

Tasks Requiring the Primary Console

Backup and Restore: Backup and restore must be run on the primary console. When the restore is completed, you must restart Tomcat on all secondary consoles.

Enter the following command:

```
/etc/init.d/novell-ac restart
```

For more information about backup and restore, see Chapter 24, “Back Up and Restore,” on page 925.

Tasks Available from the Secondary Console

When the primary console goes down, the secondary console can be used for the following tasks:

- Administrators can make configuration changes on a secondary console, and these changes are sent to Access Manager components.
- Access Manager Appliance components can use the secondary console to access their configuration information and to respond to configuration changes. When the primary console becomes functional, components revert to using the primary console, but they continue to accept commands from the secondary consoles.

Configuration Tips for the L4 Switch

When you use an L4 switch to cluster the Identity Servers, Access Gateways, or both, you need to configure it and the DNS server for each cluster. You need to configure the DNS server to resolve the base URL of the Identity Server configuration to the Identity Server VIP on the L4 switch. You need to configure the DNS server to resolve the published DNS names of the Access Gateway to the Access Gateway VIPs on the L4 switch.

In addition to this basic setup, consider the following:

- “Sticky Bit” on page 774
- “Network Configuration Requirements” on page 774
- “Health Checks” on page 775
- “Real Server Settings Example” on page 778
- “Virtual Server Settings Example” on page 779

Sticky Bit

Each L4 switch has a slightly different method and terminology for the sticky bit or persistence bind. This bit allows a client that has established a session to be directed to the same Identity Server or Access Gateway for all requests sent during the session. This minimizes the need to forward session information between Access Gateways or between Identity Servers and thus maximizes performance.

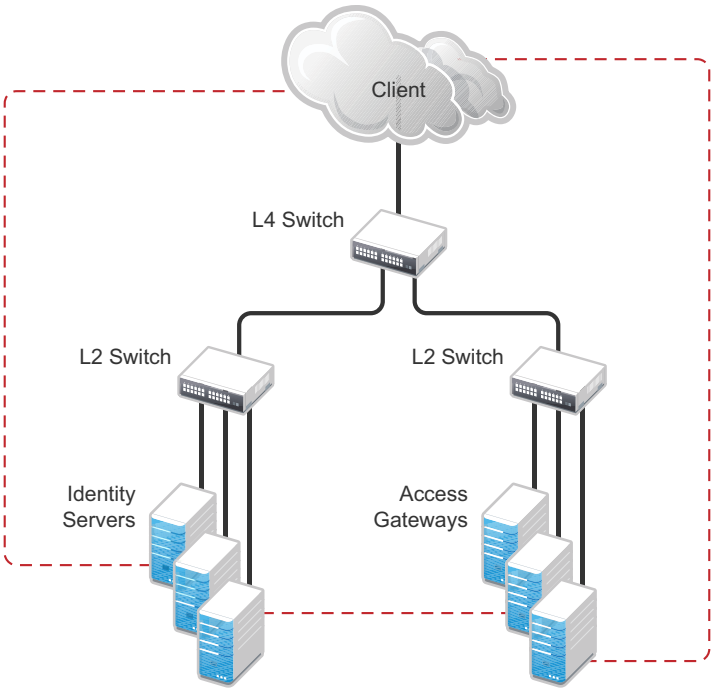
Network Configuration Requirements

When you set up the L4 switch, the following configurations are required to route all Access Manager traffic through the L4 switch:

Switches: When you install an L4 switch, you can plug the machines directly into the L4 switch or plug them into an inner switch that is plugged into the L4 switch. When you use inner switches with an L4 switch, you must use at least two inner switches: one for the Identity Servers and one for the Access Gateways. An Identity Server and an Access Gateway cannot share the same inner switch. Such a configuration causes communication problems because the Access Gateway and the Identity Server try to establish direct communication with each other rather than routing all traffic through the L4 switch.

Network Routing Requirements: You need to analyze your routing configuration. The Identity Servers and the Access Gateways must be connected to separate ports in the L4 switch. If there is a connection in your network that allows an Identity Server or an Access Gateway to communicate directly with a client without going through the L4 switch, the Access Gateway and the Identity Server try to establish direct communication with the client because networking protocols are configured to select the most direct route. Such a configuration causes communication problems because all traffic must be routed through the L4 switch. Figure 11-4 illustrates this problem.

Figure 8-1 *Network Configuration with a Potential Communication Problem*



If your network allows for this type of communication, you need to block the communication channels illustrated with the dotted lines.

Figure 11-5 shows each cluster type with its own L2 switch. An Access Gateway cluster and an Identity Server cluster cannot share the same L2 switch because they can see the MAC address for each other. Networking protocols are configured to use the most direct route for the communication, and the MAC address is more direct than going up to the L4 switch and back down. Such a configuration causes communication problems because all traffic between the clusters needs to be routed through the L4 switch. Using a separate L2 switch for each cluster type prevents them from gaining access to the MAC address and forces communication to take place through the L4 switch.

Health Checks

L4 switches use health checks to determine which cluster members are ready to receive requests and which cluster members are unhealthy and must not receive requests. You need to configure the L4 switch to monitor the heartbeat URL of the Identity Servers and Access Gateways, so that the L4 switch can use this information to update the health status of each cluster member.

The procedure is slightly different for the Identity Servers and Access Gateways:

- “Health Checks for the Identity Server” on page 776
- “Health Checks for the Access Gateway” on page 776

Health Checks for the Identity Server

The Administration Console uses the heartbeat URL to display the health status of the Identity Servers. The Identity Server heartbeat is the DNS name of the Identity Server plus the following path:

```
/nidp/app/heartbeat
```

L4 switches require you to use IP address rather than the DNS name. If the IP address of the Identity Server is 10.10.16.50, and you have configured the Identity Server for HTTPS, the heartbeat has the following URL:

```
https://10.10.16.50:8443/nidp/app/heartbeat
```

You need to configure the L4 switch to use this heartbeat to perform a health check. If you have configured SSL on the Identity Servers and your L4 switch has the ability to do an SSL L7 health check, you can use HTTPS. The SSL L7 health check returns a value of `200 OK`, indicating that everything is healthy; any other status code indicates an unhealthy state.

For a Foundry switch, the L7 health check script string must look similar to the following when the hostname is nidp1 and the IP address is 10.10.16.50:

```
healthck nidp1ssl tcp
  dest-ip 10.10.16.50
  port ssl
  protocol ssl
  protocol ssl url "GET /nidp/app/heartbeat HTTP/1.1\r\nHost: st160.lab.tst"
  protocol ssl status-code 200 200
  l7-check
```

If your switch does not support an SSL L7 health check, the HTTPS URL returns an error, usually a 404 error. Because the Identity Server heartbeat URL listens on both HTTPS and HTTP, you can use an HTTP URL for switches that do not support the SSL L7 health check. For example:

```
http://10.10.16.50:8080/nidp/app/heartbeat
```

An Alteon switch does not support the L7 health check, so the string for the health check must look similar to the following:

```
open 8080,tcp
send GET /nidp/app/heartbeat HTTP/1.1\r\nHOST:heartbeat.lab.tst \r\n\r\n
expect HTTP/1.1 200
close
```

Health Checks for the Access Gateway

External communication to the Access Gateway is typically configured to use HTTPS. In an HTTPS configuration, an L4 switch performs health checks of the Access Gateways with the published DNS name of the Access Gateway plus the following path:

```
/nesp/app/heartbeat
```

L4 switches require you to use IP address rather than the DNS name. If the IP address of the Access Gateway is 10.10.16.172, and you have configured the Access Gateway for HTTPS, the heartbeat has the following URL:

```
https://10.10.16.172:443/nesp/app/heartbeat
```

For an L4 switch to support an HTTPS query for the health of the Access Gateway, the switch must support an L7 health check. For a Foundry switch, the L7 health check script string must look similar to the following when the hostname is ag1 and the IP address is 10.10.172.

```
healthck ag1ssl tcp
  dest-ip 10.10.16.172
  port ssl
  protocol ssl
  protocol ssl url "GET /nesp/app/heartbeat HTTP/1.1\r\nHost: st160.lab.tst"
  protocol ssl status-code 200 200
  l7-check
```

If your L4 switch does not support an SSL L7 health check, the HTTPS health check URL returns an error, usually a 404 error. To solve this problem, you can create a specialized reverse proxy that opens a non-SSL port for the heartbeat URL. The following instructions configure this reverse proxy to use port 81, because port 80 on the specified IP address is reserved for redirects to the SSL port.

To create a reverse proxy for the health check:

1. In the Administration Console, click **Devices** > **Access Gateways** > **Edit** > **Reverse Proxy / Authentication**.
2. To create an additional reverse proxy service (such as *heartbeat*), click **New**, then specify a name.
3. Change the **Non-Secure Port** to 81.

   Configure the Access Gateway to listen on the same IP address as the service using port 443. For non-SSL, you must use port 81. Do not use port 80.

   For proper heartbeat information when there are multiple IP addresses configured in your Access Gateway, ensure that you configure the reverse proxy service created for the heartbeat URL to listen in the same IP address as the authenticating reverse proxy service.
4. Click **New** to create the proxy service.
5. Configure the following fields:

   Proxy Service Name: Specify a name that identifies the purpose of this proxy service.

   Published DNS Name: Specify a second DNS name that resolves to the VIP of the Access Gateways on the L4 switch. For example, if the DNS name is jwilson.provo.novell.com for the Access Gateways, you could use heartbeat.jwilson.provo.novell.com for the second name.

   Web Server IP Address: Specify the internal address:127.0.0.1.

   Host Header: Select **Forward Received Host Name**. This field is not used.
6. Click **OK**.
7. On the Reverse Proxy page, click the new proxy service, then click **Web Servers**.
8. Change the **Connect Port** value on the Web Servers page to 9009.

   The service provider (ESP) in the Access Gateway that provides the heartbeat service listens on 127.0.0.1:9009.
9. Click **Protected Resources**.
10. Click **New**, then specify a name.
11. In the URL Path List, click /*, and modify the path to contain the following value:

    ```
    /nesp/app/heartbeat
    ```

    This is the path to the heartbeat application.
12. Click **OK** > OK.

    The heartbeat of this Access Gateway is available from the following URL (See Step 4.):

    ```
    http://heartbeat.jwilson.provo.novell.com:81/nesp/app/heartbeat
    ```

If the protected resource is configured with a path of / or /*, the solution works but it can be vulnerable to attacks because the configuration opens ESP over a non-SSL port. Restricting the resource to `/nesp/app/heartbeat` automatically denies access to ESP except for the heartbeat.

13 Click **OK** and apply the changes to the configuration.

14 Add a line similar to the health check script:

For a Foundry switch, your string must look similar to the following if the hostname is ag1 and the IP address is 10.10.16.172:

```
healthck ag1 tcp
  dest-ip 10.10.16.172
  port http
  protocol http
  protocol http url "GET /nesp/app/heartbeat HTTP/1.1\r\nHost:st160.lab.tst"
  protocol http status-code 200 200
  l7-check
```

For an Alteon switch, your string must look similar to the following if the hostname is ag1 and the IP address is 10.10.16.172:

```
open 81,tcp
send GET /nesp/app/heartbeat HTTP/1.1\r\nHOST:heartbeat.lab. tst\r\n\r\n
expect HTTP/1.1 200
close
```

Real Server Settings Example

After setting up the health checks, you need to configure the real server settings. The following is an example from a Foundry switch.

```
Current real servers settings:
  1: 149.44.171.116, enabled, name 152, weight 1, timeout 10 mins, maxcon 200000
    backup none, inter 2, retry 4, restr 8
    remote disabled, proxy enabled, subnac disabled
    cookie assignment server: disabled
    exclusionary string matching: disabled
    service ports: 8443 8080
    real ports:
      8443: vport 8443, group 1, pbind clientip
          virtual server:  1, 149.44.174.220,  enabled
      8080: vport 8080, group 1, pbind clientip
          virtual server:  1, 149.44.174.220,  enabled
  2: 149.44.174.51, enabled, name brie, weight 1, timeout 10 mins, maxcon 200000
    backup none, inter 2, retry 4, restr 8
    remote disabled, proxy enabled, subnac disabled
    cookie assignment server: disabled
    exclusionary string matching: disabled
    service ports: 8443 8080
    real ports:
      8443: vport 8443, group 1, pbind clientip
          virtual server:  1, 149.44.174.220,  enabled
      8080: vport 8080, group 1, pbind clientip
          virtual server:  1, 149.44.174.220,  enabled
```

Virtual Server Settings Example

After setting up the real server settings, you need to configure the virtual server settings. The following is an example from a Foundry switch.

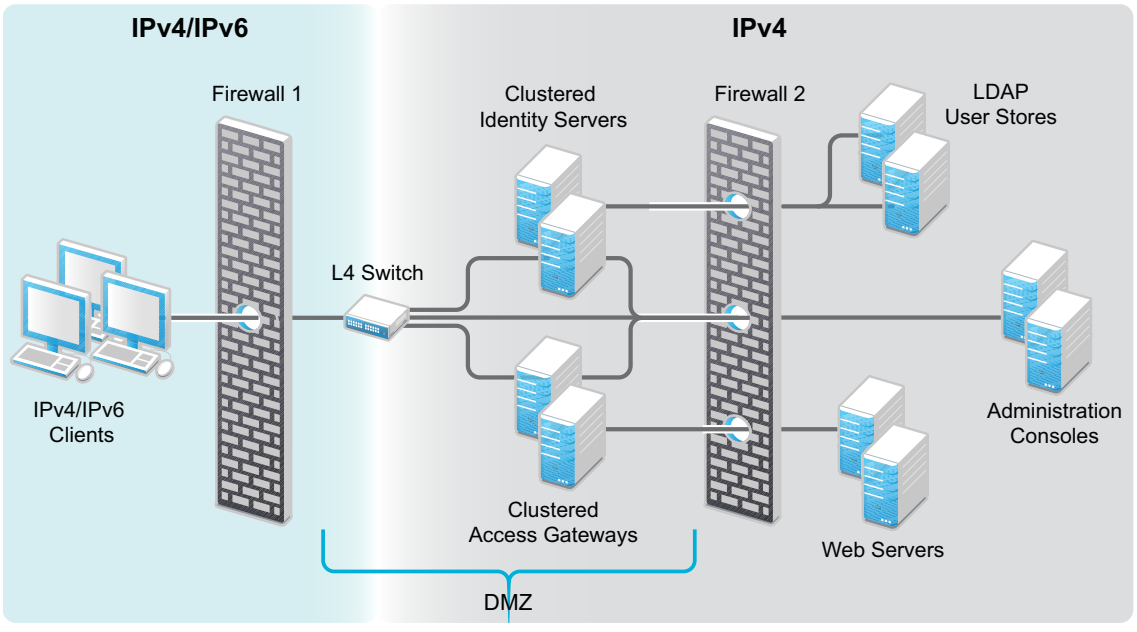
```
Current virtual servers settings:
  1: 149.44.174.220, enabled, dname idp
    virtual ports:
      8443: rport 8443, group 1, pbind clientip, frags
          real servers:
             1: 149.44.171.116,  weight 1,  enabled, backup none
             2: 149.44.174.51,   weight 1,  enabled, backup none
      8080: rport 8080, group 1, pbind clientip, frags
          real servers:
             1: 149.44.171.116,  weight 1,  enabled, backup none
             2: 149.44.174.51,   weight 1,  enabled, backup none
```

Setting up L4 Switch for IPv6 Support

IPv6 provides large number of available addresses, increased security, and reliability. For supporting IPv6, the L4 switch must support addressing both IPv4 and IPv6 as seen in the following figure.

There is no change in listener configuration for Access Manager devices. Hence, real IP configuration and real server group configuration in L4 switch remain the same. Change is required only in the virtual server configuration to accept IPv6 connections.

Figure 8-2 *L4 Switch Support For Both IPv4/IPv6*



The existing IPv4 virtual server configuration is as follows:

- server virtual vir-ipv4 10.50.50.1
- predictor round-robin
- port 8080

- port 8443
- bind 8080 real1 8080 real2 8080 real3 8080 real4 8080
- bind 8443 real1 8443 real2 8443 real3 8443 real4 8443

Use the following procedure to configure the L4 switch for IPv6 support.

1 Add an IPv6 virtual server for each of the corresponding IPv4 virtual servers:
   - server virtual vir-ipv6 2001::1
   - predictor round-robin
   - port 8080
   - port 8443
   - bind 8080 real1 8080 real2 8080 real3 8080 real4 8080
   - bind 8443 real1 8443 real2 8443 real3 8443 real4 8443

   NOTE: All real server configurations (real1, real2, real3, real4) in the above sample remains same for both IPv4 virtual server and IPv6 virtual server configurations.

2 (Optional) To use a client IPv6 address for Authorization or Identity Injection Policies, L4 switch must be configured to send X-Forwarded-For IP header with each HTTP request.

IPv6 support is explained in the following scenarios. For simplicity IPv4 information is not provided. These scenarios support IPv6 in addition to IPv4. For more information on limitations for this support, see “Limitations” on page 782.

Web SSO Over IPv6

Configuration: The L4 switch is configured to listen to the IPv6 Virtual IP addresses for both the Access Gateway and Identity Server clusters, for example, called IDP-v6 and AG-v6. The Identity Server and Access Gateway Servers must be configured in the L4 switch for listening to IPv6 requests as actual server groups IDP-Group and AG-Group. These groups serve the requests coming to IPv6 addresses configured in L4 switch.

The whole traffic to IDP-v6 and AG-v6 is forwarded to the Identity Server and Access Gateway clusters respectively with the source IP changed to the IP address of the L4 switch (IPv4-Internal).

How it works: Incoming traffic to the IDP-v6 and AG-v6 will be redirected to the IDP-Group and AG-Group based on load balancing algorithm configured in the L4 switch. The outgoing response traffic from the Identity Server and the Access Gateway Servers to the IPv6 clients will be first routed to IPv4-Internal and forwarded back to the client with source IP address of IDP-v6 and AG-v6. The traffic initiated from the Identity Servers to the Access Gateway Servers and vice versa for metadata exchange, artifact resolution and so on must also be routed through the L4 switch. Hence, the Identity Server and the Access Gateway Servers must resolve the Identity Server and the Access Gateway URL to the IPv4 addresses respectively as they understand only IPv4 addresses.

For example, if an internal DNS Server is used, then the DNS Server must be configured to resolve the Identity Server/Access Gateway Server URL. If the IPv4 address for the Identity Server is 10.75.75.1 and the Identity Server URL is www.idp.com, then the Identity Server clusters must have 10.75.75.1 www.idp.com in its hosts file.

The incoming traffic can be classified into the following:

- Traffic initiated from IPv6 clients.
- Connections initiated from the Access Gateway servers to the Identity servers.

However, both these can be considered the same as the responses from the Identity Server and the Access Gateway Servers will be using IPv4 address. The L4 switch converts the source to IPv6 address and forwards it to the respective remote parties. The clients can either be configured with IPv4 address or IPv6 address or both (dual stack). If the client is configured to use IPv6 address only or dual stack, it must resolve the published DNS names of the Identity Server and the Access Gateway Server to the IPv6 addresses respectively.

Federated SSO over IPv6

HTTP Browser clients coming in with IPv6 source address and published DNS names for Identity Provider and Service Provider URLs are accessible using IPv6 addresses. There are two ways you can access these, the Artifact or Post binding. For more information about these, see “Configuring a SAML 2.0 Profile” on page 422, “Configuring a SAML 1.1 Profile” on page 452, and “Configuring a Liberty Profile” on page 456.

Federated SSO over IPv6 Using Artifact Binding

- “Configuration” on page 781
- “How it Works?” on page 781

Configuration

The L4 switch is listening in to the IPv6 Virtual IP addresses for the Identity Server cluster. Let us call it as IDP-v6. The IPv4-Internal in the L4 switch is connected to the actual Identity Server cluster. IDP-v6 listens to IPv6 clients. The whole traffic to the IDP-v6 will be forwarded to the Identity Servers with the source IP changed to IPv4-Internal. The Identity Servers listen on the IPv4 addresses only. These IPv4 addresses of the Identity Servers must be configured as real server group, say IDP-Group in the L4 switch. This group must serve the requests coming to IDP-v6 address configured in the L4 switch. Incoming traffic to the IDP-v6 addresses will be redirected to the IDP-Group based on the load balancing algorithm configured in the L4 switch.

In case of IDP Servers acting as a Service Provider in an Artifact binding scenario, it needs to resolve the Artifact received from the Identity Provider. Hence, the Service Provider must directly contact the remote Identity Provider. There will be traffic initiated from the Service Provider in federated SSO using Artifact binding. The L4 switch needs another IPv6 interface (IPv6-Internal) to forward connections from IPv6 addresses of the Identity Servers to IPv6 addresses of remote Identity Providers. The Identity Server acting as Service Provider must be configured to contain both IPv4 and IPv6 addresses. This facilitates communication with the IPv6 address of the L4 switch. If the Identity Server is acting as an Identity Provider, there is no connection initiated from the Identity Server even in the artifact binding scenario. Hence, an internal IPv6 interface in the L4 switch is not required.

How it Works?

The outgoing response traffic from the Identity Servers to the IPv6 clients will be first routed to IPv4-Internal and forwarded back to the clients with source IP address as IDP-v6 address.

When an Identity Server is acting as a Service Provider, the traffic will be initiated from the internal Identity Servers to the remote Identity Providers. This is routed through the L4 switch and the Identity Servers must resolve the remote Identity Provider URL to the remote IPv6 address. The DNS server configured for the Identity Server must be configured to resolve the Identity Provider URL to the remote IPv6 address.

When the Identity server is acting as an Identity Provider, the incoming traffic to this Identity Server can be classified into the following:

- Traffic initiated from IPv6 clients.
- Traffic from the remote Service Provider.

However, the response from the Identity Server uses IPv4 address in both cases. L4 switch converts the response to IPv6 address and forwards it to remote IPv6 clients and Service Providers respectively.The clients can either be configured with IPv4 address or IPv6 address or both (dual stack). If the client is configured to use IPv6 address only or dual stack, it must resolve the published DNS name of the Identity Server to IDP-v6 address.

Federated SSO over IPv6 using Post Binding

- "Configuration" on page 782
- "How it Works?" on page 782

Configuration

The L4 switch is listening in to the IPv6 Virtual IP addresses for the Identity Server cluster. Let us call it as IDP-v6. The IPv4-Internal in the L4 switch is connected to the actual Identity Server cluster. IDP-v6 listens to IPv6 clients. The traffic to the IDP-v6 will be forwarded to the Identity Servers with the source IP changed to IPv4-Internal.

The Identity Servers listen on the IPv4 addresses only. These IPv4 addresses of the Identity Servers must be configured as real server group, say IDP-Group in the L4 switch. This group must serve the requests coming to IDP-v6 address configured in the L4 switch. Incoming traffic to the IDP-v6 addresses will be redirected to the IDP-Group based on the load balancing algorithm configured in the L4 switch.

Since there is no traffic initiated from the Identity Provider or Service Provider in federated SSO using Post binding, the Identity Servers must listen only using IPv4 address.

How it Works?

The outgoing response traffic from the Identity Servers to the IPv6 clients will be first routed to IPv4-Internal and forwarded back to the clients with source IP address as IDP-v6 address.

Since it is Post profile only incoming traffic will be from IPv6 clients. The clients can either be configured with IPv4 address or IPv6 address or both (dual stack). If the client is configured to use IPv6 address only or dual stack, it must resolve the published DNS name of IDP to IDP-v6 address.

Limitations

The following scenarios are not supported:

- Access Gateways communicating over IPv6 to the Web Servers listening in IPv6 addresses.
- Identity Servers communicating over IPv6 to the LDAP User stores listening in IPv6 addresses.

Using a Software Load Balancer

Instead of using an L4 switch, you can cluster the Identity Servers and the Access Gateways behind a software load balancer that runs in Layer 7. Each manufacturer uses slightly different terminology, but the basic steps are quite similar. You need to create the following types of objects:

- Pools to specify how load balancing occurs, such as round robin.
- Persistence classes to be used within the pools to enable the sticky bit or to keep state so that a connection is sent to the same device.
- Monitors to be used within the pools for monitoring the health heartbeat of the device.
- Virtual servers to set up the ports and protocols for the pools.
- Traffic IP groups where the virtual IP addresses are set up and tied to the virtual servers.

Because the software actually runs in Layer 7, it does not require any special networking setup and it runs on standard server hardware.

As an example, the following instructions explain how to configure the Zeus ZXTM Load Balancer with HTTP and HTTPS for the Identity Server and Access Gateway. For more information about this product, see Zeus Technology (http://www.zeus.com/).

1. Create two persistence classes, one for HTTPS and one for HTTP.

   ```
   HTTP > J2EE Session Persistence
   HTTPS > SSL Session ID
   ```

2. Create four monitors, two for the Identity Servers and two for the Access Gateways.

   2a. Use the following paths to specify a path for HTTP and a path for HTTPS:

   Identity Server: `/nidp/app/heartbeat`

   Access Gateway: `/nesp/app/heartbeat`

   2b. Configure the following parameters for the monitors:

   HTTP: timeout=10 seconds, use_ssl=no, host_header: <domain>, body_regex: Success

   HTTPS: timeout=10 seconds, use_ssl=yes, host_header: <domain>, body_regex: Success

   Replace <*domain*> with the DNS name of the Access Manager device

3. Create four pools, one for each monitor. Configure each pool with the following parameters:

   ```
   Load _balancing: Round Robin
   persistence: <new class created>
   max_reply_time: 10
   ```

   For an HTTP resource, replace <*new class created*> with the HTTP class you created. For an HTTPS resource replace <*new class created*> with the HTTPS class you created.

4. Create four virtual servers, one for each port. Configure each with the following parameters:

   ```
   Protocol: <scheme>
   Port: <port>
   Pool: <pool created>
   ```

   Replace <*scheme*> with HTTP or HTTPS.

   Replace <*port*> with one of the following values: 80,8080,443, or 8443.

   Replace <*pool created*> with one of the pools you created in Step 3.

5. Create two traffic manager groups, one for the Identity Servers and one for the Access Gateway.

This is where the virtual IP address is set up.

6 Start the traffic groups.

Security

Access Manager Appliance includes a certificate management service, which allows you to manage the certificates used for digital signatures and data encryption. You can create locally signed certificates or import externally signed certificates, then assign these certificates to the trust stores and keystores of the following components:

- **Identity Server:** Certificates allow you to provide secure authentication to the Identity Server and enable encrypted content from the Identity Server portal through HTTPS. They also provide secure communications between trusted Identity Servers and user stores.
- **Access Gateway:** Uses server certificates and trusted roots to protect Web servers, provide single sign-on, and enable the product's data confidentiality features, such as encryption.

You can install and distribute certificates to Access Manager Appliance components and configure how the components use certificates. This includes central storage, distribution, and expired certificate renewal.

- Chapter 9, “Securing Access Manager,” on page 787
- Chapter 10, “Understanding Access Manager Certificates,” on page 797
- Chapter 11, “Creating Certificates,” on page 799
- Chapter 12, “Managing Certificates and Keystores,” on page 807
- Chapter 13, “Assigning Certificates to Access Manager Appliance,” on page 813
- Chapter 14, “Managing Trusted Roots and Trust Stores,” on page 815
- Chapter 15, “Enabling SSL Communication,” on page 819

9 Securing Access Manager

The Administration Console contains all the configuration information for all Access Manager Appliance components. If you federate your users with other servers, it stores configuration information about these users. You need to protect the Administration Console so that unauthorized users cannot change configuration settings or gain access to the information in the configuration store. configuration store. When you develop a security plan for Access Manager Appliance, consider the following:

- "Securing the Administration Console" on page 787
- "Protecting the Configuration Store" on page 788
- "Security Considerations for Certificates" on page 788
- "Configuring Secure Communication on the Identity Server" on page 789
- "Enabling Secure Cookies" on page 790
- "Preventing Cross-site Scripting Attacks" on page 792

Securing the Administration Console

When you look for ways to secure the Administration Console from unauthorized access, consider the following:

Admin User: The admin user you create when you install the Administration Console has all rights to the Access Manager Appliance components. We recommend that you protect this account by configuring the following features:

- **Password Restrictions:** When the admin user is created, no password restrictions are set. To ensure that the password meets your minimum security requirements, you should configure the standard eDirectory password restrictions for this account. In the Administration Console, select the **Roles and Tasks** view in the iManager header, then click **Users**. Browse to the admin user (found in the novell container), then click **Restrictions**. For configuration help, use the **Help** button.
- **Intruder Detection:** The admin user is created in the novell container. You should set up an intruder detection policy for this container. In the Administration Console, select the **Roles and Tasks** view in the iManager header, then click **Directory Administration** > **Modify Object**. Select **novell,** then click **OK**. Click **Intruder Detection**. For configuration help, use the **Help** button.
- **Multiple Administrator Accounts:** Only one admin user is created when you install Access Manager Appliance. If something happens to the user who knows the name of this user and password or if the user forgets the password, you cannot access the Administration Console. Novell recommends that you create at least one backup user and make that user security equivalent to the admin user. For instructions, see "Creating Multiple Admin Accounts" on page 40. For other considerations when you have multiple administrators, see "Managing Administrators" on page 39.

Network Configuration: You need to protect the Administration Console from Internet attacks. It should be installed behind your firewall.

Delegated Administrators: If you create delegated administrators for policy containers (see “Managing Delegated Administrators” on page 41), be aware that they have sufficient rights to implement a cross-site scripting attack using the Deny Message in an Access Gateway Authorization policy.

They are also granted rights to the LDAP server, which gives them sufficient rights to access the configuration datastore with an LDAP browser. Modifications done with an LDAP browser are not logged by Access Manager. To enable the auditing of these events, see “Activating eDirectory Auditing for LDAP Events” on page 45.

Test Certificates: When you install the Administration Console, the NAM-RP certificate is automatically generated and associated with NAM-RP Reverse Proxy (**Devices** > **Access Gateways** > **[AG-Cluster]** > **[NAM-RP]**).

Protecting the Configuration Store

The configuration store is an embedded, modified version of eDirectory. It is backed up and restored with command line options, which back up and restore the Access Manager Appliance configuration objects in the ou=accessManagerContainer.o=novell object.

You should back up the configuration store on a regular schedule, and the ZIP file created should be stored in a secure place. See Section 24, “Back Up and Restore,” on page 925.

In addition to backing up the configuration store, you should also install at least two Administration Consoles (a primary and a secondary). If the primary console goes down, the secondary console can keep the communication channels open between the various components. You can install up to three Administration Consoles. For installation information, see “Installing Secondary Versions of Access Manager Appliance” on page 771.

The configuration store should not be used for a user store.

Security Considerations for Certificates

Your security deployment plan should contain policies for the following:

- **Key size for certificates:** Access Manager Appliance ships with a CA that can create certificates with a key size of 512, 1024, 2048, or 4096. Select the maximum size supported by the applications that you are protecting with Access Manager Appliance.
- **Certificate renewal dates:** Ensure that you renew certificates before it gets expired. Your security needs might allow for a longer or shorter period.
- **Trusted certificate authorities:** Access Manager Appliance ships with a CA, and during installation of the various components, it creates and distributes certificates. For added security, you might want to replace these certificates with certificates from a well-known CA.

  NOTE: Access Manager supports SHA-256 and SHA-512 as a signing algorithm.

  For more information about how to import certificates, see “Importing a Signed Certificate” on page 805.

Configuring Secure Communication on the Identity Server

The Identity Server uses the key pairs (NAM-RP-Certificate) associated with the NAM-RP Reverse Proxy Service (**Access Manager > Devices > Access Gateway > [AG-Cluster] > NAM-RP**) for secure communication. In a production environment, you should exchange the NAM-RP-Certificate that is created at the installation time with certificate from a trusted certificate authority.

The Identity Server uses the key pair for following scenarios:

- To establish SSL communication between the Identity Server and the browsers and between the Identity Server and the Access Gateway for back-channel communications.
- To sign authentication requests, to sign communication with providers on the SOAP back channel, and to sign Web Service Provider profiles.
- To encrypt specific fields or data in the assertions. For more information about the services that use the certificate for encryption, see “Viewing Services That Use the Encryption” on page 790
- To enable secure communication between the user store and the Identity Server, you can also import the trusted root certificate of the user store. For configuration information, see “Configuring Identity User Stores” on page 272

This section describes the following tasks:

- “Viewing the Services That Use the Signing” on page 789
- “Viewing Services That Use the Encryption” on page 790

Viewing the Services That Use the Signing

The following services can be configured to use signing:

- “Protocols” on page 789
- “SOAP Back Channel” on page 790
- “Profiles” on page 790

Protocols

The protocols can be configured to sign authentication requests and responses.

To view your current configuration:

1. In the Administration Console, click **Devices** > **Identity Servers** > **Edit**.
2. In the **Identity Provider** section, view the setting for the **Require Signed Authentication Requests** option. If it is selected, all authentication requests from service providers must be signed.
3. In the **Identity Consumer** section, view the settings for the **Require Signed Assertion**s and **Sign Authentication Requests** options. If these options are selected, assertions and authentication requests are signed.

SOAP Back Channel

The SOAP back channel is the channel that the protocols use to communicate directly with a provider. The SOAP back channel is used for artifact resolutions and attribute queries for the Identity Web Services Framework.

To view your current configuration for the SOAP back channel:

1. In the Administration Console, click **Devices** > **Identity Servers** > **Edit**.
2. Select the protocol (Liberty, SAML 1.1, or SAML 2.0), then click the name of an identity provider or service provider.
3. Click **Trust**.
4. View the **Security** section. If the **Message Signing** option is selected, signing is enabled for the SOAP back channel.

Profiles

Any of the Web Service Provider profiles can be enabled for signing by configuring them to use X.509 for their message-level security mechanism.

To view your current configuration:

1. In the Administration Console, click **Devices** > **Identity Servers** > **Edit** > **Liberty** > **Web Service Provider**.
2. Click the name of a profile, then click **Descriptions**.
3. Click the **Description Name**.
4. If either **Peer entity = None, Message=X509** or **Peer entity = MutualTLS, Message=X509** has been selected as the security mechanism, signing has been enabled for the profile.

Viewing Services That Use the Encryption

All of the Liberty Web Service Provider Profiles allow you to configure them so that the resource IDs are encrypted. By default, no profile encrypts the IDs.

To view your current configuration:

1. In the Administration Console, click **Devices > Identity Servers > Edit** > **Liberty** > **Web Service Provider**.
2. Click the name of a profile.
3. If the **Have Discovery Encrypt This Service's Resource IDs** option is selected, the encryption key pair is used to encrypt the resource IDs.

Enabling Secure Cookies

The Access Gateway and the Embedded Service Provider of the Access Gateway both use session cookies in their communication with the browser. The following sections explain how to protect these cookies from being intercepted by hackers.

- "Securing the Embedded Service Provider Session Cookie on the Access Gateway" on page 791
- "Securing the Proxy Session Cookie" on page 792

For more information about making cookies secure, see the following documents:

- Secure attribute for cookies in RFC 2965 (http://www.faqs.org/rfcs/rfc2965.html)
- HTTP-only cookies (http://msdn.microsoft.com/en-us/library/ms533046.aspx)

Securing the Embedded Service Provider Session Cookie on the Access Gateway

An attacker can spoof a non-secure browser into sending a JSESSION cookie that contains a valid user session. This might happen because the Access Gateway communicates with its Embedded Service Provider on port 8080, which is a non-secure connection. Because the Embedded Service Provider does not know whether the Access Gateway is using SSL to communicate with the browsers, the Embedded Service Provider does not mark the JSESSION cookie as secure when it creates the cookie. The Access Gateway receives the Set-Cookie header from the Embedded Service Provider and passes it back to the browser, which means that there is a non-secure, clear-text cookie in the browser. If an attacker spoofs the domain of the Access Gateway, the browser sends the non-secure JSESSION cookie over a non-secure channel where the cookie might be sniffed.

To stop this, you must first configure the Access Gateway to use SSL. See “Configuring SSL Communication with Browsers and the Identity Server” on page 828. After you have SSL configured, you must configure Tomcat to secure the cookie.

1. On the Access Gateway server, log in as an admin user.
2. Change to the Tomcat configuration directory.

   `/opt/novell/nam/mag/conf/`
3. In a text editor, open the `server.xml` file.
4. Search for the connector on port 9009.
5. Add the following parameter within the `Connector` element:

   `secure="true"`
6. Save the `server.xml` file.
7. Enter one of the following commands to restart Tomcat:

   `/etc/init.d/novell-mag restart` OR `rcnovell-mag restart`

Preventing Automatically Changing Session ID

1. Go to Devices > Access Gateway > Edit > Reverse Proxy / Authentication > ESP Global Options.
2. Set **RENAME_SESSIONID** to false. By default, this is set to true.
3. Restart Tomcat on each Identity Server in the cluster.

Securing the Proxy Session Cookie

The proxy session cookies store authentication information and other information in temporary memory that is transferred between the browser and the proxy. These cookies are deleted when the browser is closed. However if these cookies are sent through a non-secure channel, there is a threat of hackers intercepting the cookies and impersonating a user on Web sites. To stop this, you can use the following configuration options:

- “Setting an Authentication Cookie with a Secure Keyword for HTTP” on page 792
- “Preventing Cross-Site Scripting Vulnerabilities” on page 792

Setting an Authentication Cookie with a Secure Keyword for HTTP

You can configure the Access Gateway to force the HTTP services to have the authentication cookie set with the keyword secure.

To enable this option:

1. In the Administration Console, click **Devices** > **Access Gateways** > **Edit** > **Reverse Proxy / Authentication**.
2. Select the **Enable Secure Cookies** option, then click **OK** twice.
3. Update the Access Gateway.

This option is used to secure the cookie when the Access Gateway is placed behind an SSL accelerator, such as the Cisco SSL accelerator, and the Access Gateway is configured to communicate by using only HTTP.

Preventing Cross-Site Scripting Vulnerabilities

Cross-site scripting vulnerabilities in Web browsers allow malicious sites to grab cookies from a vulnerable site. The goal of such attacks might be to perform session fixation or to impersonate the valid user. You can configure the Access Gateway to set its authentication cookie with the `HttpOnly` keyword, to prevent scripts from accessing the cookie.

To enable this option:

1. In the Administration Console, click **Devices** > **Access Gateways** > **Edit** > **Reverse Proxy / Authentication**.
2. Enable the **Force HTTP-Only Cookies** option, then click **OK** twice.
3. Update the Access Gateway.

Preventing Cross-site Scripting Attacks

By default, Access Manager does extensive checks to prevent Cross-site Scripting (XSS) attacks. However, Access Manager does not validate a JSP file if you have customized it. If you modify JSP files to customize the login, logout, error pages, and so forth, you must sanitize the JSP file to prevent XSS attacks.

You need to perform either one of the following options to sanitize the customized JSP file:

- “Option 1: HTML Escaping” on page 793
- “Option 2: Filtering” on page 793

Option 1: HTML Escaping

Perform the following XSS checks for the customized JSP file to protect it from possible XSS attacks. For more information about XSS prevention techniques, see XSS (Cross Site Scripting) Prevention Cheat Sheet.

Perform the following steps:

1. Verify if the `org.apache.commons.lang.StringEscapeUtils` class is available in the JSP file.

   For example, the following import statement should be available in the import section of the JSP file:

   ```
   <%@ page import="org.apache.commons.lang.StringEscapeUtils"%>
   ```

2. Verify if all URL query parameter values are sanitized.

   The following code snippet sample shows how URL query parameter values (uname and target) can be sanitized:

   ```
   <%//Fetch the values from URL query parametersString target = (String)
   request.getAttribute("target");String uname = (String)
   request.getAttribute("username"); String sanitizedUName = ""; if (uname !=
   null){//Sanitize the value assigned to uname sanitizedUName =
   StringEscapeUtils.escapeHtml(uname); } String sanitizedTarget = ""; if (target
   != null){ //Sanitize the value assigned to target query param sanitizedTarget =
   StringEscapeUtils.escapeHtml(target);}%>
   ```

3. Add double quotes ("") in value attribute (or any attribute that is dynamically assigned) for any HTML element that get assigned with above URL query param value.

   ```
   <!-- The last 2 double quotes are mandatory to prevent XSS attacks --><input
   type="text" class="smalltext" name="Ecom_User_ID" size="30"
   value="<%=sanitizedUName%>">......<!-- The last 2 double quotes are mandatory
   to prevent XSS attacks --><input type="hidden" name="target"
   value="<%=sanitizedTarget%>">
   ```

4. Restart the component whose JSP file you have modified. For example, if you modify the Identity Server's JSP file, restart the Identity Server by running the following command:

   ```
   sh /etc/init.d/novell-idp restart
   ```

Option 2: Filtering

By deafult, the XSS detection filter is enabled in Identity provider's `web.xml` file:

- **Linux**: `/opt/novell/nam/idp/webapps/nidp/WEB-INF`
- **Windows**: `C:\Program Files (x86)\Novell\Tomcat\webapps\nidp\WEB-INF`

The filter is as follows:

```
<filter>
           <filter-name>XSSDetectionFilter</filter-name>
             <filter-
class>com.novell.nidp.servlets.filters.xss.XSSDetectionFilter</filter-class>
             <description>This filter is used to detect XSS attacks in NIDS</
description>
             <init-param>
             <param-name>active</param-name>
                  <param-value>True</param-value>
             </init-param>
             <init-param>
                  <param-name>level</param-name>
                  <param-value>SCRIPT_TAGS</param-value>
            </init-param>
            <init-param>
                  <param-name>exclude</param-name>
                  <param-value>soap,wstrust,metadata,oauth</param-value>
            </init-param>
</filter>
```

To disable it, set the `<param-value> True` to False as follows:

```
<init-param>
      <param-name>active</param-name>
      <param-value>False</param-value>
</init-param>
```

To exclude it from a specific request, add a URL string from that request in the `<param-name>exclude</param-name>` tag that contains the default excluded request path name.

For example: If `wsfed` request fails due to some reason, add `wsfed` in the exclude list. Now, Identity Provider will not filter `wsfed` specific requests.

The exclude init-param is as follows:

```
<init-param>
   <param-name>exclude</param-name>
   <param-value>soap,wstrust,metadata,oauth,wsfed</param-value>
</init-param>
```

NOTE: It is recommended to use the above option as it overrides the following approach:

This approach might have a minor performance impact due to the checks it performs. If you perform HTML escaping in customized JSP pages, you do not need to perform this additional filtering.

Perform the followings steps to sanitize the Identity Server's customized JSP file:

1. Download the `eMFrame_xss.jar` file.

   This library prevents XSS based attacks.

2. Place this library at the following location:

   `/opt/novell/nids/lib/webapp/WEB-INF/lib`

3. Add a filter in the `web.xml`file located at the following location:

   `/opt/novell/nam/idp/webapps/nidp/WEB-INF`.

```
<filter><filter-name>XSS</filter-name><display-name>XSS</display-
name><description>Filters XSS injections.</description> <filter-
class>com.novell.emframe.fw.filter.CrossScriptingFilter</filter-class></
filter> <filter-mapping><filter-name>XSS</filter-name><url-pattern>/*</url-
pattern></filter-mapping>
```

4 Restart the Identity Server by running the following command:

```
sh /etc/init.d/novell-idp restart
```

10 Understanding Access Manager Certificates

Access Manager Appliance allows you to manage centrally stored certificates used for digital signatures and data encryption. eDirectory resides on the Administration Console and is the main certificate store for all of the Access Manager Appliance components. If you use a Novell Certificate Server, you can create certificates there and import them into Access Manager Appliance.

By default, all Access Manager Appliance components (Identity Server and Access Gateway ~~and SSL VPN~~) trust the local Access Manager Appliance certificate authority (CA). However, if the Identity Server is configured to use an SSL certificate signed externally, the trust store of the Embedded Service Provider for each component must be configured to trust this new CA.

Certificate management commands issued from a secondary Administration Console can work only if the primary console is also running properly. Other commands can work independently of the primary console.

You can create and distribute certificates to the following components:

- **Identity Server:** Uses certificates and trust stores to provide secure authentication to the Identity Server and enable encrypted content from the Identity Server portal via HTTPS. Certificates also provide secure communications between trusted Identity Servers and user stores.

  Liberty and SAML 2.0 protocol messages that are exchanged between identity and service providers often need to be digitally signed. The Identity Server uses the signing certificate included with the metadata of a trusted provider to validate signed messages from the trusted provider. For protocol messages to be exchanged between providers through SSL, each provider must trust the CA of the other provider. You must import the public key of the CA used by the other provider.

  The Identity Server also has a trust store for OCSP (Online Certificate Status Protocol) certificates, which is used to check the revocation status of a certificate.
- **Access Gateway:** Uses server certificates and trusted roots to protect Web servers, provide single sign-on, and enable the product's data confidentiality features, such as encryption. They are used for background communication with the Identity Server and policy engine and to establish trust between the Identity Server and the Access Gateway.

To ensure the validity of X.509 certificates, Access Manager Appliance supports both Certificate Revocation Lists (CRLs) and Online Certificate Status Protocol (OCSP) methods of verification.

When X509 authentication is configured as the authentication contract, it works even after you revoke the certificate for the X509 mutual authentication. When you access the nidp login page from the client browser and select the revoked certificate, browser does not throw an error message telling that the certificate has been revoked. You can either issue a CRL or wait until the next CRL issuance date. The revoked certificates will work until the next CRL issuance date.

If you do not want to wait and issue a CRL now, perform the following steps:

1. Navigate to **Roles and Tasks** > **NetIQ Certificate Server** > **Configure Certificate Authority** > **CRL**.
2. Click **CRL**.

3 Under Next CRL Issuance, click **Issue Now**.

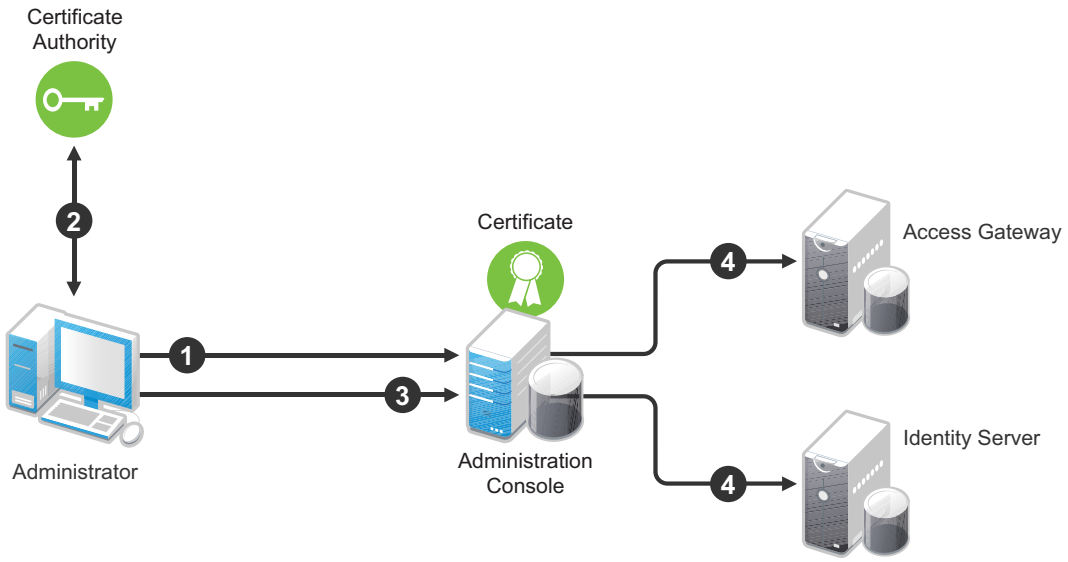
4 Click **OK**.

5 Restart the Identity Server.

Process Flow

You can install and distribute certificates to the Access Manager Appliance components and configure how the components use certificates. This includes central storage, distribution, and expired certificate renewal. Figure 10-1 illustrates the primary administrative actions for certificate management in Access Manager Appliance:

Figure 10-1 *Certificate Management*



1. Generate a certificate signing request (CSR). See “Generating a Certificate Signing Request” on page 804.
2. Send the CSR to the external certificate authority (CA) for signing.

   A CA is a third-party or network authority that issues and manages security credentials and public keys for message encryption. The CA’s certificate is held in the configuration store of the computers that trust the CA.
3. Import the signed certificate and CA chain into the configuration store. See “Importing Public Key Certificates (Trusted Roots)” on page 815.
4. Assign certificates to devices. See “Assigning Certificates to Access Manager Appliance” on page 813.

If you are unfamiliar with public key cryptography concepts, see “Public Key Cryptography Basics” in the *Novell Certificate Server 3.1.1 Guide*.

11 Creating Certificates

Access Manager Appliance comes with certificates for testing purposes. At a minimum, you must create one SSL certificates for Identity Server and Access Gateway reverse proxy (NAM-RP). Then you replace the predefined certificates with the new ones.

If you install a secondary Administration Console, the certificate authority (CA) is installed with the first instance of eDirectory, and the secondary consoles have eDirectory replicas and therefore no CA software. All certificate management must be done from the primary Administration Console. Certificate management commands issued from a secondary Administration Console can work only if the primary console is also running properly. Other commands can work independently of the primary console.

NOTE: It is recommended to recreate the certificates after upgrading Access Manager to 4.2. Certificates created in Access Manager 4.1 or earlier versions use SHA-1 ciphers by default. In Access Manager 4.2 onward, certificates are created using SHA-2 ciphers by default.

IMPORTANT: Before generating any certificates with the Administration Console CA, ensure that time is synchronized within one minute among all of your Access Manager Appliance devices. If the time of the Administration Console is ahead of the device for which you are creating the certificate, the device rejects the certificate.

---

1. In the Administration Console, click **Security** > **Certificates**.
2. Select from the following actions:

   New: To create a new certificate, click **New**. For information about the fields you need to fill in, see “Creating a Locally Signed Certificate” on page 799 and “Generating a Certificate Signing Request” on page 804.

   Delete: To delete a certificate, select the certificate, then click **Delete**. If the certificate is assigned to a keystore, a warning message appears. You must remove a certificate from all keystores before it can be deleted.

   Import Private/Public Keypair: To import a key pair, click **Import Private/Public Keypair**. For more information, see “Importing a Private/Public Key Pair” on page 812.

Creating a Locally Signed Certificate

By default, the Access Manager Appliance installation process creates the local CA that can issue and sign certificates and installs a certificate server that generates certificates, keys, and CSRs (certificate signing requests) and imports certificates and keys.

1. In the Administration Console, click **Security** > **Certificates**.
2. Click **New**.
3. Select the following option:

   Use local certificate authority: Creates a certificate signed by the local CA (or Organizational CA), and creates the private key. For information about creating a CSR, see “Generating a Certificate Signing Request” on page 804.

4 Provide a certificate name:

Certificate name: The name of the certificate. Pick a unique, system-wide name for the certificate that you can easily associate with the certificate's purpose. The name must contain only alphanumeric characters and no spaces.

5 For **Subject**, click **Edit** to display a dialog box that lets you add the appropriate attributes for the subject name.

The subject is an X.500 formatted distinguished name that identifies the entity that is bound to the public key in an X.509 certificate. Choose the subject name that the browser expects to find in the certificate. The name you enter must be fully distinguished. Completing all the fields creates a fully distinguished name that includes the appropriate types (such as C for country, ST for state, L for location, O for organization, OU for organizational unit, and CN for common name). For example, cn=AcmeWebServer.ou=Sales.o=Acme.c=US.

Common name: If you are creating a certificate for an Identity Server, specify the DNS name of the Identity Server. If you are creating a certificate for an Access Gateway, specify the published DNS name of the proxy service. Specifying values for the other attributes is optional.

For more information about the other attributes, see "Editing the Subject Name" on page 801.

6 Click **OK**, then fill in the following fields:

Signature algorithm: The algorithm you want to use (SHA-256 or SHA-512).

Valid from: The date from which the certificate is valid. For externally signed certificates, the external certificate authority sets the validity period.

Months valid: The number of months that the certificate is valid.

Key size: The size of the key. Select 512, 1024, 2048, or 4096.

7 (Optional) To configure advanced options, click **Advanced Options**.

8 Configure the following options as necessary for your organization:

Critical: Specifies that an application should reject the certificate if the application does not understand the key usage extensions.

Encrypt other keys: Specifies that the certificate is used to encrypt keys.

Encrypt data directly: Encrypts data for private transmission to the key pair owner. Only the intended receiver can read the data.

Create digital signatures: Specifies that the certificate is used to create digital signatures.

Non-repudiation: Links a digital signature to the signer and the data. This prevents others from duplicating the signature because no one else has the signer's private key. Additionally, the signer cannot deny having signed the data.

9 (Conditional) If you are creating a key for a certificate authority, configure the following options:

This key is for a Certificate Authority: Specifies that this certificate is for the local configuration (eDirectory) certificate authority.

If you create a new CA, all the keys signed by the CA being replaced no longer have a trusted CA. You might also need to reassign the new CA to all the trust stores that contained the old CA.

Critical: Enforces the basic constraints you specify. Select one of the following:

- **Unlimited:** Specifies no restriction on the number of subordinate certificates that the CA can verify.
- **Do not allow intermediate signing certificates in certificate chain:** Prevents the CA from creating other CAs, but it can create server or user certificates.
- **Number of allowable intermediate signing certificates in signing chain:** Specifies how many subordinate certificates are allowed in the certificate chain. Values must be 1 or more. Entering 0 creates only entity objects.

10 (Optional) To create subject alternative names used by the certificate, click **Edit Subject Alternate Names**, then click **New**.

Alternate names can represent the entity identified by the certificate. The certificate can identify the subject CN=www.OU=novell.O=com, but the subject can also be known by an IP address, such as 222.111.100.101, or a URI, such as www.novell.com, for example. For more information, see "Assigning Alternate Subject Names" on page 803.

11 Click **OK**.

12 (Conditional) If you assigned alternate names, determine how you want applications to handle the alternate names. Select **Critical** if you want an application that does not understand the alternate name extensions to reject the certificate.

13 Click **OK**.

Editing the Subject Name

1 Fill in one or more of the following attributes.

The following attributes are the most common ones used in certificate subjects:

Common name: The DNS name of the server.

Specify the value, for example AcmeWebServer.provo.com. Do not include the type (cn=). The UI adds that for you.

For the Identity Server, this is the domain name of the base URL of the Identity Server configuration. This value cannot be an IP address or begin with a number, in order to ensure that trust does not fail between providers.

For the Access Gateway, this is the published DNS name of the proxy service.

Organizational unit: Describes departments or divisions.

Organization: Differentiates between organizational divisions.

City or town: Commonly referred to as the Locality.

State or province: Commonly referred to as the State. Do not abbreviate the name.

Country: The country, such as US.

2 Use the drop-down menus to add additional attributes.

These values allow you to specify additional fields that are supported by eDirectory, and you can include them as part of the subject to further identify the entity represented by the certificate.

CN: The **Common name** attribute in the list of **Commonly used attributes** (OID: 2.5.4.3)

C: The **Country attribute** in the list of **Commonly used attributes** (OID: 2.5.4.6)

SN: The surname attribute (OID: 2.5.4.4)

L: The locality attribute, which is the **City or town** attribute in the list of **Commonly used attributes** (OID: 2.5.4.7)

ST: The **State or province** attribute in the list of **Commonly used attributes** (OID: 2.5.4.8)

S: The **State or province** attribute in the list of **Commonly used attributes** (OID: 2.5.4.8)

O: The Organization attribute in the list of Commonly used attributes (OID: 2.5.4.10)

OU: The Organizational unit attribute in the list of Commonly used attributes (OID: 2.5.4.11)

street: Describes the street address (OID: 2.5.4.9)

serialNumber: Specifies the serial number of a device (OID: 2.5.4.5)

title: Describes the position or function of an object (OID: 2.5.4.12)

description: Describes the associated object (OID: 2.5.4.13)

searchGuide: Specifies a search filter (OID: 2.5.4.14)

businessCategory: Describes the kind of business performed by an organization (OID: 2.5.4.15)

postalAddress: Specifies address information required for the physical delivery of postal messages (OID: 2.5.4.16)

postalCode: Specifies the postal code of an object (OID: 2.5.4.17)

postOfficeBox: Specifies the post office box for the physical delivery of mail (OID: 2.5.4.18)

physicalDeliveryOfficeName: Specifies the name of the city or place where a physical delivery office is located (OID: 2.5.4.19)

telephoneNumber: Specifies a telephone number (OID: 2.5.4.20)

telexNumber: Specifies a telex number (OID: 2.5.4.21)

teletexTerminalIdentifier: Specifies an identifier for a telex terminal (OID: 2.5.4.22)

facsimileTelephoneNumber: Specifies the telephone number for a facsimile terminal (OID: 2.5.4.23)

x121Address: Specifies the address used in electronic data exchange (OID: 2.5.4.24)

internationalISDNNumber: Specifies an international ISDN number used in voice, video, and data transmission (OID: 2.5.4.25)

registeredAddress: Specifies the postal address for the delivery of telegrams or expedited documents (OID: 2.5.4.26)

destinationIndicator: Specifies an attribute used in telegram services (OID: 2.5.4.27)

preferredDeliveryMethod: Specifies the preferred delivery method for a message (OID: 2.5.4.28)

presentationAddress: Specifies an OSI presentation layer address (OID: 2.5.4.29)

supportedApplicationContext: Specifies the identifiers for the OSI application contexts in the application layer (OID: 2.5.4.30)

member: Specifies the distinguished name of an object associated with a group or a list (OID: 2.5.4.31)

owner: Specifies the name of an object that has responsibility for another object (OID: 2.5.4.32)

roleOccupant: Specifies the distinguished name of an object that fulfills an organizational role (OID: 2.5.4.33)

seeAlso: Specifies the distinguished name of an object that contains additional information about the same real-world object (OID: 2.5.4.34)

userPassword: Specifies the object's password (OID: 2.5.4.35)

name: Specifies a name that is in the UTF-8 form of the ISO 10646 character set (OID: 2.5.4.41)

givenName: Specifies the given or first name of an object (OID: 2.5.4.42)

initials: Specifies the initials of an object (OID: 2.5.4.43)

generationQualifier: Specifies the generation of an object, which is usually a suffix (OID: 2.5.4.44)

x500UniqueIdentifier: Specifies an identifier that distinguishes between objects when a DN has been reused (OID: 2.5.4.45)

dnQualifier: Specifies information that makes an object unique when information is being merged from multiple sources and objects could have the same RDNs (OID: 2.5.4.46)

enhancedSearchGuide: Specifies a search filter used by X.500 users (OID: 2.5.4.47)

protocolInformation: Specifies information that is used with the presentationAddress attribute (OID: 2.5.4.48)

distinguishedName: Specifies the distinguished name of an object (OID: 2.5.4.49)

uniqueMember: Specifies the distinguished name of an object associated with a group or a list (OID: 2.5.4.50)

houseIdentifier: Identifies a building within a location (OID: 2.5.4.51)

dmdName: Specifies a directory management domain (OID: 2.5.4.54)

E: Specifies an e-mail address.

EM: Specifies an e-mail address.

DC: Specifies the domain name for an object (OID: 0.9.2342.19200300.100.1.25)

uniqueID: Contains an RDN-type name that can be used to create a unique name in the tree (OID: 0.9.2342.19200300.100.1.1)

T: Specifies the name of the tree root object (OID: 2.16.840.1.113719.1.1.4.1.181)

OID: Specifies an object identifier in dot notation.

3 To create a certificate, continue with Step 6 on page 800, or to create a signing request, continue with Step 5 on page 804.

Assigning Alternate Subject Names

1 Fill in the following fields:

Name Type: Names as specified by RFC 2459. Use the drop-down list to specify a name type, such as:

- **Directory name:** An X.500 directory name. The required format for the name is `.<attribute name>=<attribute value>`. For example:

  ```
  .O=novell.C=US
  ```

  Access Manager Appliance supports the following attributes:

  Country (C)
  Organization (O)
  Organizational Unit (OU)
  State or Province (S or ST)
  Locality (L)
  Common Name (CN)
- **IP Address:** An IP address such as 222.123.123.123
- **URI:** A URI such as www.novell.com.
- **Registered ID:** An ASN.1 object identifier.
- **DNS Name:** A domain name such as novell.com.
- **Email Address (RFC 822 name):** An e-mail address such as ca@novell.com.
- **X400 Name:** The messaging and e-mail standard specified by the ITU-TS (International Telecommunications Union - Telecommunication Standard Sector). It is an alternative to the more prevalent Simple Mail Transfer Protocol (SMTP) e-mail protocol. X.400 is common in Europe and Canada.
- **EDI Party:** EDI (Electronic Data Interchange) is a standard format for exchanging business data.

- **Other:** A user-defined name.

Name: The display alternative name.

2 Continue with Step 11 on page 801.

Generating a Certificate Signing Request

1 In the Administration Console, click **Security** > **Certificates** > **New**.

2 To create a certificate signing request (CSR), select **Use external certificate authority**.

This option generates a CSR for you to send to the CA for signing. A third-party CA is managed by a third party outside of the eDirectory tree. An example of a third party CA is VeriSign. After the signed certificate is received, you need to import the certificate.

3 Specify a Certificate name.

Pick a unique, system-wide name for the certificate that you can easily associate with the certificate's purpose. The name must contain only alphanumeric characters and no spaces.

4 Click the **Edit** button to display a dialog box that lets you add appropriate locality information types for the subject name.

For more information, see "Editing the Subject Name" on page 801.

5 Click **OK**, then fill in the following fields:

Signature algorithm: The algorithm you want to use (SHA-256 or SHA-512).

Valid from: The date from which the certificate is valid. For externally signed certificates, the external certificate authority sets the validity period.

Months valid: The number of months that the certificate is valid.

Key size: The size of the key. Select 512, 1024, 2048, or 4096.

6 (Conditional) If you are creating a key for a certificate authority, click **Advanced Options**, then configure the following:

This key is for a Certificate Authority: Select this option.

Critical: Enforces the basic constraints you specify. Select one of the following:

- **Unlimited:** Specifies no restriction on the number of subordinate certificates that the CA can verify.
- **Do not allow intermediate signing certificates in certificate chain:** Prevents the CA from creating other CAs, but it can create server or user certificates.
- **Number of allowable intermediate signing certificates in signing chain:** Specifies how many subordinate certificates are allowed in the certificate chain. Values must be 1 or more. Entering 0 creates only entity objects.

7 Click **OK**.

8 Click the name of the certificate, copy the CSR data and send the information to the external CA.

The certificate status is CSR Pending until you import the signed certificate.

9 Click **Close**.

10 When you receive the signed certificate and the trusted root (CA chain), continue with "Importing a Signed Certificate" on page 805.

Importing a Signed Certificate

After you receive the signed certificate and the CA chain, you must import it. CA can return the certificate in multiple ways. Typically, the CA either returns one or more files each containing one certificate, or returns a file with multiple certificates in it.

The following figure illustrates a certificate chain example.

Figure 11-1 *Illustration of a Certificate Chain Example*



To import this certificate chain:

1. In the Administration Console, click **Security** > **Certificates**, then click the name of a certificate that is in a CSR Pending state.
2. Click **Import Signed Certificate**.
3. In the Import Signed Certificate dialog box, browse to locate the Entity certificate data file or paste the Entity certificate data text into the **Certificate data text** field.
4. To import the CA chain, click **Add trusted root** and then locate the Root certificate data.
5. Click **Add intermediate certificate** if you need to continue adding certificates to the chain for example, add Intermediate cert 1 and cert 2 in that order.
6. Click **OK**, then click **Close** on the Certificate Details page.

The certificate is now available for use by Access Manager Appliance devices.

NOTE: When there is a server certificate and more than two intermediate CA certificates, use PKCS7 format file and import the certificate and its CA chain.

If you receive an error when attempting to import the certificate, see Chapter , “Troubleshooting Certificate Issues,” on page 1000.

12 Managing Certificates and Keystores

You can import certificates created by an external certificate authority. These certificates then need to be assigned to a device by adding the certificate to the device's keystore. The subject name of the certificate needs to match the DNS name of the device, or if you are using wildcard certificates, the main domain name needs to match. You can perform the following certificate tasks:

- "Viewing Certificate Details" on page 807
- "Renewing a Certificate" on page 809
- "Exporting a Private/Public Key Pair" on page 811
- "Exporting a Public Certificate" on page 811
- "Importing a Private/Public Key Pair" on page 812

Viewing Certificate Details

The Certificate Details page lists the properties of a certificate, such as certificate type, name, subject, and assigned keystores. The fields are not editable.

1. In the Administration Console, click **Security** > **Certificates**.
2. Select one of the following:
   - Click the name of a certificate that is not in a CSR Pending state. The Certificate Details page contains the following information about the certificate:

Field	Description
Issuer	The name of the CA that created the certificate.
Serial number	The serial number of the certificate.
Subject	The subject name of the certificate.
Valid from	The first date and time that the certificate is valid.
Valid to	The date and time that the certificate expires.
Devices	The devices that are configured to hold this certificate on their file system and the keystore that holds them.
Key size	The key size that was used to create the certificate.
Signature algorithm	The signature algorithm that was used to create the certificate.
Finger print (MD5)	The certificate's message digest that was calculated with the MD5 algorithm. It is embedded into the certificate at creation time. It can be used to uniquely identify a certificate. For example, users can verify that a certificate is the one they think it is by matching this published MD5 fingerprint with the MD5 fingerprint on the local certificate.

Field	Description
Finger print (SHA256)	The certificate's message digest that was calculated with the SHA-256 algorithm. It is embedded into the certificate at creation time. It can be used to uniquely identify a certificate. For example, users can verify that a certificate is the one they think it is by matching a published SHA-256 fingerprint with the SHA-256 fingerprint on the local certificate.
Subject Alternate Names: Critical	Indicates whether an application should reject the certificate if the application does not understand the alternate name extensions. Any configured alternate names are displayed in the list.
Key Usage: Critical	Indicates whether an application should reject the certificate if the application does not understand the key usage extensions.
Sign CRLs	Indicates whether the certificate is used to sign CRLs (Certificate Revocation Lists).
Sign certificates	Indicates whether the certificate is used to sign other certificates.
Encrypt other keys	Indicates whether the certificate is used to encrypt keys.
Encrypt data directly	Indicates whether the certificate can encrypted data for private transmission to the key pair owner. Only the intended receiver can read the data.
Create digital signatures	Indicates whether the certificate can create digital signatures.
Non-repudiation	Indicates whether the certificate links a digital signature to the signer and the data. This prevents others from duplicating the signature because no one else has the signer’s private key. Additionally, the signer cannot deny having signed the data.
CRL Distribution Points	A list of Certificate Revocation List (CRL) distribution points that are embedded into the certificate as an extension at certificate creation time. Implementations search the CRL from each distribution point (the distribution point is usually a URI that points to a store of revoked certificates) to see whether a certificate has been revoked.
Authority Info Access (OCSP)	A list of Online Certificate Status Protocol (OCSP) responders that are embedded into the certificate as an extension at certificate creation time. Implementations query the OCSP responder to see whether a certificate has been revoked.

- Click the name of a certification in a CSR Pending state. The following information is displayed:

Subject	The subject name of the certificate.
Valid from	The date and time that the request was generated.
Valid to	The date and time that the request expires.
Devices	No entries. A CSR cannot be assigned to a device.
Key size	The key size that was used to create the request.
Signature algorithm	The signature algorithm that was used to create the request.
State	Displays `CSR Pending`, indicating that the request has been generated.
CSR data	The certificate signing request data. You can either export this data or copy and paste it into CA's request tool.

3 (Conditional) For a certificate not in a CSR Pending state, select one of the following actions:

Renew: Allows you to renew the certificate. For more information, see "Renewing a Certificate" on page 809.

Export Private/Public Keypair: Allows you to export private certificates to obtain a backup copy of the key, to move the key to a different server, or to share the key between servers. For more information, see "Exporting a Private/Public Key Pair" on page 811

Export Public Certificate: Allows you to export a public key certificate to a file. For more information, see "Exporting a Public Certificate" on page 811.

4 (Conditional) For a certificate in a CSR Pending state, select one of the following actions:

Import Signed Certificate: Allows you to import the certificate that was generated for this request. For more information, see "Importing a Signed Certificate" on page 805.

Export CSR: Allows you to export the CSR to a CSR file.

NOTE: Whenever the configuration store contains a Key Material Object (KMO) with a CSR in pending state, the KMO will not be exported by using the `amdiagcfg` script and not be backed up by using the `ambkup` script.

Renewing a Certificate

The Certificate Details page lists the properties of a certificate, such as certificate type, name, subject, and assigned keystores. This page also includes the original CSR when the certificate is still in a pending state (for example, you have generated the CSR, but you have not yet received and imported the signed certificate). If the certificate is expiring, you can cut and paste its text to send it to the CA to get a renewed certificate, then import the newly signed certificate.

For the certificates that Access Manager Appliance uses internally, a certificate process is started with Tomcat. This process runs once every 24 hours. It checks all the internal certificates and determines if they are going to expire within 30 days. If they are due to expire, the process automatically regenerates the certificate or trusted root. When a certificate is regenerated, the following message appears:

```
One or more automatically created certificates were regenerated. Reboot the entire
administration console as soon as possible to avoid interruption of service.
```

This message appears when the administrator logs in to the Administration Console, or if the administrator is already logged in, when the administrator switches from one page to another.

This event is also auditing. Another audit event is also generated which tells the administrator to restart any effected services. When the Administration Console certificate and the eDirectory certificates are expiring, a log entry is written to the app_sc log file. The log entry contains the “Recreating auto-generated certificates” string as well as a couple success or failure messages per key re-generated.

Certificates and trusted roots that are manually created with the Access Manager Appliance CA or are imported into Administration Console use a different process. The administrator is warned that these certificates are expiring when the administrator logs in to the Administration Console. The following message is displayed:

`Warning: the following certificates are expired or will expire within X days: <certA>, <certB>.`

This message is displayed each time the administrator logs in to the Administration Console. Events for the expiration of these certificates are not audited and are not logged.

The following figure illustrates the certificate chain example.

Figure 12-1 *Illustration of a Certificate Chain Example*



To renew a certificate:

1. In the Administration Console, click **Security** > **Certificates**.
2. Click the certificate name.
3. Click **Renew**.
4. On the Renew page, either browse to locate and select the certificate or select the **Certificate data text (PCM/Base64)** option and paste the certificate data into the text box.
5. To import the CA chain, click **Add trusted root** and then locate the Root certificate data.
6. Update the device using the certificate.
7. Click **Add intermediate certificate** if you need to continue adding certificates to the chain for example, add Intermediate cert 1 and cert 2 in that order.
8. Click **OK**, then click **Close**.

Exporting a Private/Public Key Pair

When you create a certificate, you can specify whether it is exportable. If a key is exportable, it can be extracted and put in a file along with the associated certificate. The file is written in an industry standard format, PKCS#12, which allows it to be transported to other platforms. It is encrypted with a user-specified password to protect the private key. You can export private certificates to obtain a backup copy of the key, to move the key to a different server, or to share the key between servers.

You cannot export a certificate if you enabled the **Do not allow private key to be exportable option** while creating the certificate.

1. In the Administration Console, click **Security** > **Certificates**.
2. On the Certificates page, click the certificate.
3. On the Certificate Details page, click **Export Private/Public Keypair**.
4. Select a format for the key:

   PFX/PKCS12: Public Key Cryptography Standards #12 (PKCS#12) format, which is also called PFX format. This format can be used to create JKS or PEM files.

   JKS: Java keystore format.
5. Specify the password in the **Encryption/decryption** password field, then click OK.

   IMPORTANT: Remember this password because you need it to re-import the key.
6. Click **OK**.

Exporting a Public Certificate

You can export a trusted root or a public key certificate to a file so that a client can use it to verify the certificate chain sent by a cryptography-enabled application, or to have a backup copy of the file.

You can export the certificate in the following formats:

- DER-encoded (`.der`) to a file.
- PEM-encoded to a file. This is a Base64-encoded DER certificate that is enclosed between the BEGIN CERTIFICATE and END CERTIFICATE tags.
- PEM CUT/Paste Buffer. This displays the certificate data so you can copy it to the system Clipboard. You can then pasted it directly into a cryptography-enabled application.

To export the public certificate:

1. In the Administration Console, click **Security** > **Certificates**.
2. Click the certificate name.
3. On the Certificate Details page, click **Export Public Certificate**, then click the file type.
4. Save the output file to the location of your choosing.

Importing a Private/Public Key Pair

If you created a key pair that was exported from another certificate management system, you can import the key pair and then assign it to an Access Manager device. The file needs to be in PFX/PKCS12 (*.pfx or *.p12) format.

1. In the Administration Console, click **Security** > **Certificates**.
2. Choose **Actions > Import Private/Public Keypair**.
3. Fill in the following fields:

   Certificate name: The name of the certificate. This is a system-wide, unique name used by Access Manager. The name must contain only alphanumeric characters and no spaces. If the name starts with a number, an underline (_) prefix is added to the name so that the name conforms to XML requirements. If the name contains invalid characters, it is automatically renamed.

   Keystore password: Type the encryption/decryption password established when exporting the certificate.

   Certificate data file (PFX/PKCS12): The certificate file to import. You can browse to locate the *.pfx or *.p12 file.

   Certificate data file (JKS): To locate a JKS file, select this option, then click **Browse**.
4. Click **OK**.

   If you receive an error when importing the certificate, the error comes from either NICI or PKI. For a description of these error codes, see *Novell Certificate Server Error Codes and Novell International Cryptographic Infrastructure* (http://www.novell.com/documentation/nwec/index.html). For general certificate import issues, see “Importing an External Certificate Key Pair” on page 1001.

13 Assigning Certificates to Access Manager Appliance

This section discusses how you update, renew, and assign certificates to Access Manager Appliance.

The Access Gateway can be configured to use certificates for SSL communication with two types of entities:

- **Client Browsers:** You can enable SSL communication between the client browsers and the Access Gateway. When setting up this feature, you can either have the Access Manager Appliance CA automatically generate a certificate key or you can select a certificate key you have already imported (or created) for the reverse proxy. To manage this certificate in the Administration Console, click **Access Gateways** > **[Configuration Link]** > **[Name of Reverse Proxy]**. For more information, see “Managing Reverse Proxies and Authentication” on page 94.
- **Protected Web Servers:** You can enable SSL communication between the Access Gateway and the protected Web servers. This option is only available if you have enabled SSL communication between the browsers and the Access Gateway. You can enable SSL or mutual SSL. To manage these certificates in the Administration Console, click **Access Gateways** > **[Configuration Link]** > **[Name of Reverse Proxy]** > **[Name of Proxy Service]** > **Web Servers**.

14 Managing Trusted Roots and Trust Stores

- "Managing Trusted Roots" on page 815
- "Viewing External Trusted Roots" on page 817

Managing Trusted Roots

A certificate from a certificate authority (CA) is commonly referred to as trusted root. A trusted root is a trusted certificate, or the certificate of a known CA. These certificates are self-signed and are recognized as representing a CA that is trusted. To validate a digital signature, you must trust at least one of the certificates in the user or server's certificate chain. You can directly trust the certificate of the user or server, or you can choose to trust any other certificate in the chain. Typically, the certificate that is trusted is the root CA's certificate.

1. In the Administration Console, click **Security** > **Trusted Roots**.
2. Select form the following actions:

   Import: Allows you to import trusted roots so that Access Manager devices can trust the certificate sent by other computers at runtime. For more information, see "Importing Public Key Certificates (Trusted Roots)" on page 815.

   Delete: To delete a trusted root, select the trusted root, then click **Delete**.

   Auto Import From Server: To import a trusted root from another server, click **Auto Import From Server**. For more information, see "Auto-Importing Certificates from Servers" on page 816.

Importing Public Key Certificates (Trusted Roots)

You import trusted roots so that the specific device can trust the certificate sent by other computers at runtime. After you import a trusted root, you can assign it to the proper trust store associated with a device, which allows the device to trust certificates signed by the trusted root.

1. In the Administration Console, click **Security** > **Trusted Roots**.
2. Click **Import**, then specify a name for the certificate.

   This is a system-wide, unique name used by Access Manager Appliance.
3. Select one of the following methods to import the public key:
   - **Certificate data file (DER/PEM/PKCS7):** Select this method to browse to a file. Click **Browse** to locate the file on your file system.
   - **Certificate data text (PEM/Base64):** Select this method to paste Base64-encoded certificate data text.
4. Click **OK**.

Auto-Importing Certificates from Servers

You can import certificates from other servers (such as an LDAP server, an identity provider, or service provider) and make them available for use in Access Manager Appliance. You must provide the IP address, port, and certificate name.

1. In the Administration Console, click **Security** > **Trusted Roots > Auto-Import from Server**.
2. Fill in the following fields:

   Server IP Address: Specify the server IP address. You can use a DNS name.

   Server Port: Specify the server port.

   Certificate Name: Specify a unique name of the certificate to store in Access Manager.
3. Click **OK**.

Exporting the Public Certificate of a Trusted Root

You can export a trusted root or a public key certificate to a file so that a client can use it to verify the certificate chain sent by a cryptography-enabled application, or to have a backup copy of the file.

You can export the certificate in the following formats:

- DER-encoded (`.der`) to a file.
- PEM-encoded to a file. This is a Base64-encoded DER certificate that is enclosed between BEGIN CERTIFICATE and END CERTIFICATE tags.
- PEM CUT/Paste Buffer. This displays the certificate data so you can copy it to the system Clipboard. You can then pasted it directly into a cryptography-enabled application.

To export the public certificate:

1. In the Administration Console, click **Security** > **Trusted Roots**.
2. Click the name of the trusted root.
3. On the Certificate Details page, click **Export Public Certificate**, then click the file type.
4. Save the output file.

Viewing Trusted Root Details

1. In the Administration Console, click **Security** > **Trusted Roots**.
2. Click the name of a trusted root.
3. View the following information:

Field	Description
Issuer	The name of the CA that created the certificate.
Serial number	The serial number of the certificate.
Subject	The subject name of the certificate.
Valid from	The first date and time that the certificate is valid.
Valid to	The date and time that the certificate expires.
Key size	The key size that was used to create the certificate.

Field	Description
Signature algorithm	The signature algorithm that was used to create the certificate.
Finger print (MD5)	The certificate's message digest that was calculated with the MD5 algorithm. It is embedded into the certificate at creation time. It can be used to uniquely identify a certificate. For example, users can verify that a certificate is the one they think it is by matching this published MD5 fingerprint with the MD5 fingerprint on the local certificate.
Finger print (SHA256)	The certificate's message digest that was calculated with the SHA-256 algorithm. It is embedded into the certificate at creation time. It can be used to uniquely identify a certificate. For example, users can verify that a certificate is the one they think it is by matching a published SHA-256 fingerprint with the SHA-256 fingerprint on the local certificate.

The **Subject Alternate Names** section indicates whether an application should reject the certificate if the application does not understand the alternate name extensions. Any configured alternate names are displayed in the list.

The **Key Usage** section indicates whether an application should reject the certificate if the application does not understand the key usage extensions. The following are possible:

Sign CRLs: Indicates whether the certificate is used to sign CRLs (Certificate Revocation Lists).

Sign certificates: Indicates that the certificate is used to sign other certificates.

Encrypt other keys: Indicates that the certificate is used to encrypt keys.

Encrypt data directly: Indicates that the certificate encrypts data for private transmission to the key pair owner. Only the intended receiver can read the data.

Create digital signatures: Indicates that the certificate is used to create digital signatures.

Non-repudiation: Indicates that the certificate links a digital signature to the signer and the data. This prevents others from duplicating the signature because no one else has the signer's private key. Additionally, the signer cannot deny having signed the data.

CRL Distribution Points: Displays a list of Certificate Revocation List (CRL) distribution points that are embedded into the certificate as an extension at certificate creation time. Implementations search the CRL from each distribution point (the distribution point is usually a URI that points to a store of revoked certificates) to see whether a certificate has been revoked.

Authority Info Access (OCSP): Displays a list of Online Certificate Status Protocol (OCSP) responders that are embedded into the certificate as an extension at certificate creation time. Implementations query the OCSP responder to see whether a certificate has been revoked.

4 **Export Public Certificate:** Allows you to export a trusted root to a file so that a client can use it to verify the certificate chain sent by a cryptography-enabled application. For more information, see "Exporting a Public Certificate" on page 811.

5 Click **Close**.

Viewing External Trusted Roots

The Identity Server uses local Access Manager Appliance CA and external certificate authorities to verify the SSL certificates. The external certificates are listed in the **External Trusted Roots** tab.

NOTE: All the well-known trusted roots are added to the proxy trust store during the Access Manager Appliance Installation.

1 In Administration Console, click **Security** > **Trusted Roots** > **External Trusted Roots**.

The **External Trusted Roots** tab lists all the external trusted roots that Access Manager Appliance supports.

2 View the following information:

Field	Description
Alias	The name of the certificate as seen by the Access Manager appliance.
Issuer	The name of the CA that created the certificate.
Subject	The subject name of the certificate.
Starting Date	The date and time from which the certificate is valid.
Ending Date	The date and time till that the certificate is valid.

15 Enabling SSL Communication

- “Enabling SSL Communication” on page 819
- “Using SSL on the Access Manager Appliance Communication Channels” on page 826
- “Prerequisites for SSL” on page 827
- “Configuring SSL Communication with Browsers and the Identity Server” on page 828
- “Configuring SSL between the Proxy Service and the Web Servers” on page 828
- “Configuring the SSL Communication” on page 829

Enabling SSL Communication

NetIQ Access Manager Appliance enables SSL communication with the Default Reverse Proxy and the Identity Server, using a self signed certificate.

You can configure the Access Gateway to use SSL in its connections to the browsers, and to its Web servers.

- “Using Access Manager Certificates” on page 819
- “Using Externally Signed Certificates” on page 822
- “SSL Renegotiation” on page 825

Using Access Manager Certificates

However, the browsers are not set up to trust the Access Manager CA. You need to import the public key of the trusted root certificate (configCA) into the browsers to establish the trust.

Configuring the Access Gateway for SSL

This section describes how to set up SSL for the Access Gateway communication channels:

Configuring SSL Communication with the Browsers and the Access Gateway

1. In the Administration Console, click **Devices** > **Access Gateways** > **Edit** > **[Name of Reverse Proxy]**.
2. To configure the reverse proxy for SSL, fill in the following fields:

   Enable SSL with Embedded Service Provider: Select this option to encrypt the data exchanged for authentication (the communication channel between the Identity Server and the Access Gateway). This option is only available for the reverse proxy that has been assigned to perform authentication.

   If you enable SSL between the browsers and the Access Gateway, this option is automatically selected for you. You can enable SSL with the embedded service provider without enabling SSL between the Access Gateway and the browsers. This allows the authentication and identity

information that the Access Gateway and the Identity Server exchange to use a secure channel, but allows the Access Gateways to use non-secure channels with the browsers and the Web servers. This saves processing overhead if the data on the Web servers is not sensitive.

Enable SSL between Browser and Access Gateway: Select this option to require SSL connections between your clients and the Access Gateway. SSL must be configured between the browsers and the Access Gateway before you can configure SSL between the Access Gateway and the Web servers. For this process, see "Enabling SSL between the Reverse Proxy and Its Web Servers" on page 821.

Redirect Requests from Non-Secure Port to Secure Port: Determines whether browsers are redirected to the secure port and allowed to establish an SSL connection. If this option is not selected, browsers that connect to the non-secure port are denied service.

3 Generate a certificate key by using the Access Manager CA:

3a Click **Auto-generate Key**, then click **OK** twice.

3b On the Select Certificate page, make sure the certificate is selected, then click **OK**.

The generated certificate appears in the **Server Certificate** text box.

4 Configure the ports for SSL:

Non-Secure Port: Specifies the port on which to listen for HTTP requests. The default port for HTTP is 80. If you have selected the **Redirect Requests from Non-Secure Port to Secure Port** option, requests sent to this port are redirected to the secure port. If the browser can establish an SSL connection, the session continues on the secure port. If the browser cannot establish an SSL connection, the session is terminated.

Secure Port: Specifies the port on which to listen for HTTPS requests (which is usually 443). This port needs to match the configuration for SSL. If SSL is enabled, this port is used for all communication with the browsers. The listening address and port combination must not match any combination you have configured for another reverse proxy or tunnel.

5 In the **Proxy Service List**, click **[Name of Proxy Service]** > **Protected Resources**.

6 In the **Protected Resource List**, change the Authentication Procedure from an HTTP contract to an HTTPS contract.

For example, if a protected resource is using the **Name/Password - Basic** contract, click the name and change it to the **Name/Password - Form**, the **Secure Name/Password - Basic** or the **Secure Name/Password - Form** contract. Then click **OK**.

The **Name/Password - Form** contract is capable of using either HTTP or HTTPS.

To enable single sign-on, select the same contract for all the protected resources.

7 Click the **Configuration Panel** link near the bottom of the page, then in the confirmation box, click **OK**.

8 On the Server Configuration page, click **Reverse Proxy / Authentication**.

9 In the **Embedded Service Provider** section, click **Auto-Import Identity Server Configuration Trusted Root**, click **OK**, specify an alias, click **OK** twice, then click **Close**.

This option imports the public key of the Identity Server into the trust store of the embedded service provider. This sets up a trusted SSL relationship between the embedded service provider and the Identity Server.

The configCA public key certificate of the Access Manager CA is automatically added to the ESP Trust Store. If you are using Access Manager CA certificates for the Identity Server, you do not need to import the configCA certificate unless someone has deleted it from this trust store.

10 Click **Configuration Panel**, then in the confirmation box, click **OK**.

11 On the Server Configuration page, click **OK**.

12 On the Access Gateways page, click **Update** > **OK**.

13 Update the Identity Server so that it uses the new SSL configuration. Click **Devices** > **Identity Servers**, then click **Update** > **OK**.

14 Verify that the trusted relationship between the Identity Server and the Access Gateway has been reestablished:

14a Enter the URL to a protected resource on the Access Gateway. For example, enter

```
https://www.mytest.com
```

14b Complete one of the following:

- If you are prompted for login credentials, enter them. The trusted relationship has been reestablished.
- If you receive a 100101043 or 100101044 error, the trusted relationship has not been established. For information about solving this problem, see "Troubleshooting 100101043 and 100101044 Liberty Metadata Load Errors" on page 982.

Enabling SSL between the Reverse Proxy and Its Web Servers

To enable SSL between the reverse proxy and the Web servers, you must have already performed the following tasks:

❒ Enabled SSL between the Access Gateway and the browsers. See "Configuring a Reverse Proxy" on page 85 and select the **Enable SSL between Browser and Access Gateway** field.

❒ Enabled SSL on the Web server. See your Web server documentation.

If you have completed these tasks:

1 In the Administration Console, click **Devices** > **Access Gateways** > **Edit** > **[Name of Reverse Proxy]** > **[Name of Proxy Service]** > **Web Servers**.

The Web Servers configuration page appears.

2 To configure SSL, select **Connect Using SSL**.

This option is not available if you have not set up SSL between the browsers and the Access Gateway. See "Configuring a Reverse Proxy" on page 85 and select the **Enable SSL between Browser and Access Gateway** field.

3 In the **Connect Port** field, specify the port that your Web server uses for SSL communication.

4 Configure how you want the certificate verified. The Access Gateway supports different options. Select one of the following:

- **Do not verify:** Select this option if you do not want to verify the **Web Server Trusted Root** certificate. Continue with Step 3.
- To verify the certificate authority of the Web server certificate, select **Any in Reverse Proxy Trust Store**. When this option is selected, the public certificate of the certificate authority must be added to the proxy trust store.

IMPORTANT: For an Access Gateway Service, this option is a global option. If you select this option for one proxy service, all proxy services on an Access Gateway Service are flagged to verify the public certificate. This verification is done even when other proxy services are set to **Do not verify**.

---

5 Click the **Manage Reverse Proxy Trust Store** icon. The auto import screen appears.

6 Ensure that the IP address of the Web server and the port match your Web server configuration.

If these values are wrong, you have entered them incorrectly on the Web server page. Click **Cancel** and reconfigure them before continuing.

7 Click **OK**.

Wait while the Access Gateway retrieves the server certificate, the root CA certificate, and any CA certificates from a chain from the Web server.

8 Specify an alias, then click **OK**.

All the displayed certificates are added to the trust store.

9 Click **Close**.

10 (Optional) For mutual authentication:

10a Select the certificate. Click the **Select Certificate** icon, select the certificate you created for the reverse proxy, then click **OK**.

10b Import the trusted root certificate of the CA that signed the proxy service's certificate to the Web servers assigned to this proxy service.

See your Web server documentation for instructions.

11 Click **Configuration Panel**, then click **OK**.

12 On the **Configuration** page, click **OK**.

13 On the **Access Gateways** page, click **Update**.

14 (Optional). Test this configuration from a client browser:

14a Enter the published DNS name as the URL in the browser.

14b Click the links that require authentication for access.

Using Externally Signed Certificates

When the Identity Server is configured to use an SSL certificate that is signed externally, the trusted store of the embedded service provider for each component must be configured to trust this new CA. The browsers that are used to authenticate to the Identity Server must be configured to trust the CA that created the certificate for the Identity Server. If you obtain a certificate from a well-known external CA, most browsers are already configured to trust certificates from well-known CAs.

The following procedures explain how to use certificates signed by an external Certificate Authority.

- "Obtaining Externally Signed Certificates" on page 822
- "Configuring the Access Gateway to Use an Externally Signed Certificate" on page 825

Obtaining Externally Signed Certificates

The following sections explain how to create certificate signing requests for the Identity Server and Access Gateway, how to use the requests to obtain signed certificates, then how to import the signed certificates and the root certificate of the Certificate Authority into Access Manager Appliance.

- "Creating the Certificate Signing Request" on page 822
- "Getting a Signed Certificate" on page 823
- "Importing the Signed Certificates and Root Certificate" on page 824

Creating the Certificate Signing Request

You need to create two certificate signing requests: one for the Identity Server and one for the Access Gateway. The **Certificate name** and the **Common name** need to be different, but the other values can be the same.

What you need to know or create	Example	Your Value
Certificate name	ipda_test or lag_test	_____
Certificate Subject Fields:		
Common name	ipda.test.novell.com or lag.test.novell.com	_____
Organizational unit	novell	_____
Organization	test	_____
City or town	Provo	_____
State or province	UTAH	_____
Country	US	

To create a signing request for the Identity Server:

1. In the Administration Console, click **Security** > **Certificates** > **New**.
2. Select the **Use External certificate authority** option.
3. Fill the following fields:

   Certificate name: `idpa_test`

   Signature algorithm: Accept the default.

   Valid from: Accept the default.

   Months valid: Accept the default.

   Key size: Accept the default.
4. Click the **Edit** icon on the **Subject** line.
5. Fill in the following fields:

   Common name: `idpa.test.novell.com`

   Organizational unit: `novell`

   Organization: `test`

   City or town: `Provo`

   State or province: `UTAH`

   Country: `US`
6. Click **OK** twice, then click the name of the certificate.
7. Click **Export CSR**.

   The signing request is saved to a file.
8. Repeat Step 1 through Step 7 to create a signing request for the Access Gateway.

Getting a Signed Certificate

You can send the certificate signing request to a certificate authority and wait for the CA to return a signed certificate or you can use a trial certificate for testing while you wait for the official certificate. Companies such as VeriSign offer trial signed certificates for testing.

Modify the following instructions for the CA you have selected to sign your certificates:

1. Set up an account with a certificate authority and select the free trial option.
2. Open your certificate signing request for the Identity Server in a text editor.
3. Copy and paste the text of the certificate request into the appropriate box for a trial certificate.
4. If CA requires that you select a server platform, select eDirectory if available. If eDirectory is not a choice, select unknown or server not listed.
5. Click **Next**, then copy the signed certificate and paste it into a new text file or at the bottom of the signing request file.
6. Click **Back**, and repeat Step 2 through Step 5 for the Access Gateway.
7. Follow the instructions of the vendor to download the root certificate of the Certificate Authority and any intermediate CA certificates.

Importing the Signed Certificates and Root Certificate

The following steps explain how to imported the signed certificates and the trust root into the Administration Console so that they are available to be assigned to key stores and trusted root stores.

1. In the Administration Console, click **Security** > **Trusted Roots**.
2. Click **Import**, then specify a name for the root certificate.
3. Either click **Browse** and locate the root certificate file or select **Certificate data text** and paste the certificate in the text box.
4. Click **OK**.

   The trusted root is added and is now available to add to trusted root stores.
5. (Conditional) Repeat Step 2 through Step 4 for any intermediate CA certificates.
6. In a text editor, open the signed certificate for the Identity Server.
7. In the Administration Console, click **Security** > **Certificates**, then click the name of certificate signing request for the Identity Server.
8. Click **Import Signed Certificate**, then select **Certificate data text (PEM/Based64)**.
9. Paste the text for the signed certificate into the data text box. Copy everything from

   ```
   -----BEGIN CERTIFICATE-----
   ```

   through

   ```
   -----END CERTIFICATE-----
   ```
10. Click **Add trusted root**, then either click **Browse** and locate the root certificate file or select **Certificate data text** and paste the certificate in the text box.
11. (Conditional) For any intermediate CA certificates, click **Add intermediate certificate**, then either click **Browse** and locate the intermediate certificate file or select **Certificate data text** and paste the certificate in the text box.
12. Click **OK**.

    The certificate is now available to be assigned to the keystore of a device.

    If the certificate fails to import and you receive an error, it is probably missing a trusted root certificate in a chain of trusted roots. To determine whether this is the problem, see “Resolving a -1226 PKI Error” on page 1001 and “Importing an External Certificate Key Pair” on page 1001.
13. Repeat Step 6 through Step 12 for the Access Gateway certificate.

Configuring the Access Gateway to Use an Externally Signed Certificate

This section explains how to enable SSL communication between the Access Gateway and the Identity Server and between the Access Gateway and the browsers.

1. In the Administration Console, click **Devices** > **Access Gateways** > **Edit** > **[Name of Reverse Proxy]**.
2. Select **Enable SSL between Browser and Access Gateway**.
3. In the **Server Certificate** line, click the **Browse** icon.
4. Select the Access Gateway certificate, then click **OK**.

   IMPORTANT: If the external certificate authority writes the DN in reverse order (the cn element comes first rather than last), you receive an error message that the subject name does not contain the cn of the device. You can ignore this warning, if the order of the DN elements is the cause.
5. Specify an **Alias** for the certificate, then click **OK** > **Close**.
6. On the Reverse Proxy page, click **OK**.
7. On the Server Configuration page, click **Reverse Proxy / Authentication**.
8. Click **OK** twice to return to the Access Gateways page.
9. On the Access Gateways page, click **Update**.
10. Verify that the trusted relationship between the Identity Server and the Access Gateway has been reestablished:

    10a Enter the URL to a protected resource on the Access Gateway.

    10b Complete one of the following:

    - If you are prompted for login credentials, enter them. The trusted relationship has been reestablished.
    - If you receive a 100101043 or 100101044 error, the trusted relationship has not been established. For information about solving this problem, "Troubleshooting 100101043 and 100101044 Liberty Metadata Load Errors" on page 982.

SSL Renegotiation

SSL renegotiation is the process of establishing a new SSL handshake over an existing SSL connection. SSL renegotiation can be initiated either by the SSL client or the SSL server. Initiating an SSL renegotiation on the client or the server requires different set of APIs. The renegotiation messages (ciphers and encryption keys) are encrypted and then sent over the existing SSL connection to establish another session securely and is useful in the following scenarios:

- When you require a client authentication.
- When you require a different set of encryption and decryption keys.
- When you require a different set of encryption and hashing algorithms.

SSL renegotiation is enabled or disabled by the following parameter:
`"sun.security.ssl.allowUnsafeRenegotiation.`

NOTE: By default this parameter is disabled.

This is defined in a registry on Windows and a configuration file on SLES.

You can verify whether the Identity Server, Access Gateway and Administration Console support secure renegotiation by using the following command:

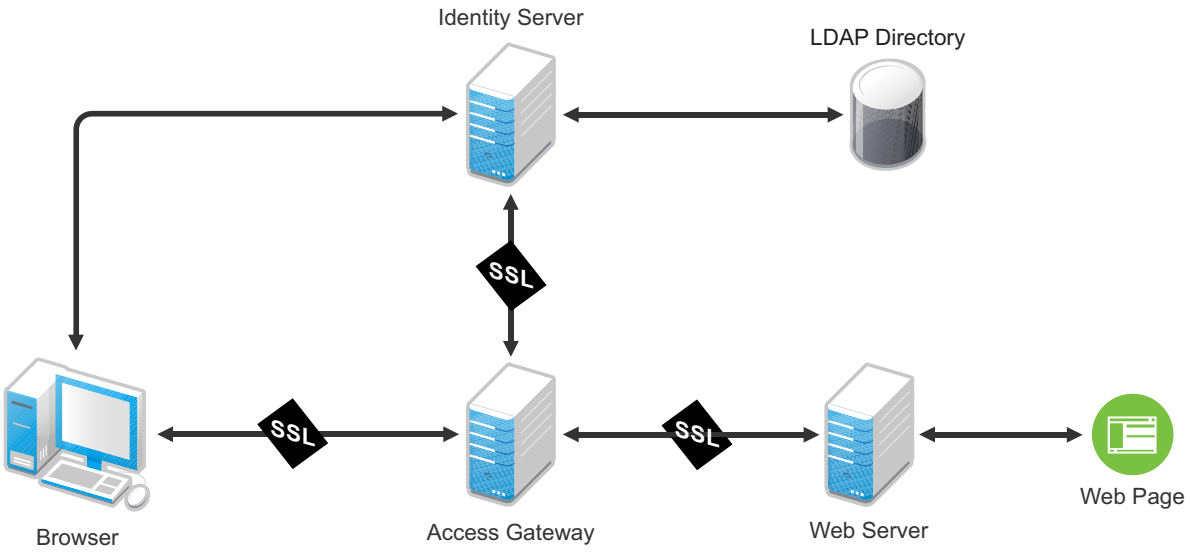
```
openssl s_client -connect <IP address of the Access Manager component:port>
```

Port can either be 8443 or 443 based on the Access Gateway configuration.

Using SSL on the Access Manager Appliance Communication Channels

You can configure the Access Manager Appliance to use SSL in its connections to the Identity Server, to the browsers, and to its Web servers. Figure 15-1 illustrates these communication channels.

Figure 15-1 *Setting Up SSL for the Access Gateway Communication Channels*



This section only describes how to set up SSL for the Access Gateway communication channels. The Identity Server needs to be configured for SSL before the Access Gateway can be configured for SSL.

When a user logs in to the Identity Server, the Identity Server verifies the user's credentials, usually with the credentials stored in an LDAP directory, but other methods are available. If the login is successful, the Identity Server sends an artifact to the browser, and the browser forwards it to the Access Gateway. The Access Gateway uses the artifact to retrieve the user's name and password from the Identity Server. The Access Gateway and Identity Server channel is probably the first communication channel you should enable for SSL. The Access Gateway uses an Embedded Service Provider to communicate with the Identity Server. When you enable SSL between the two, the Access Manager distributes the necessary certificates to set up SSL. However, if you have configured the Identity Server to use certificates from an external certificate authority (CA), you need to import the public certificate of this CA into the trust store of the Access Gateway. If you have set up the Access Gateway to use a certificate from an external CA, you need to import the public certificate of this CA into the trust store of the Identity Server.

SSL must be enabled between the Access Gateway and the browsers before you can enable SSL between the Access Gateway and its Web servers. If you enable SSL between the Access Gateway and the browsers, SSL is automatically enabled for the Access Gateway Embedded Service Provider

that communicates with the Identity Server. After you have enabled SSL between the Access Gateway and the browsers, you can select whether to enable SSL between the Access Gateway and the Web servers. By not enabling SSL to the Web servers, you can save processing overhead if the data on the Web servers is not sensitive or if it is already sufficiently protected.

Whether you need the added security of SSL or mutual SSL between the Access Gateway and its Web servers depends upon how you have set up your Web servers.

- You should enable at least SSL if the Access Gateway is injecting authentication credentials into HTTP headers.
- Mutual SSL is probably not needed if you have configured the Web servers so that they can only accept connections with the Access Gateway.

Prerequisites for SSL

The following SSL configuration instructions assume that you have already created or imported the certificate that you are going to use for SSL. This certificate must have a subject name (cn) that matches the published DNS name of the proxy service that you are going to use for authentication. You can obtain this certificate one of two ways:

- You can use the Access Manager CA to create this certificate. See “Creating a Locally Signed Certificate” on page 799.
- You can create a certificate signing request (CSR), send it to an external CA, then import the returned certificates into Access Manager. See “Generating a Certificate Signing Request” on page 804 and “Importing Public Key Certificates (Trusted Roots)” on page 815.

Prerequisites for SSL Communication between the Identity Server and Access Manager Appliance

If you are going to set up SSL communication between the Identity Server and the Access Gateway for authentication and you have configured the Identity Server to use certificates created by an external CA, you need to import the public certificate of this CA into the trusted root keystore of the Access Gateway.

1. If you have not imported the public certificate of this CA into the trusted root store of the Identity Server, do so now. For more information, see “Importing Public Key Certificates (Trusted Roots)” on page 815.
2. To add the public certificate to the Access Gateway:
   - **2a** In the Administration Console, click **Devices** > **Access Gateways** > **Edit** > **Service Provider Certificates** > **Trusted Roots**
   - **2b** In the Trusted Roots section, click **Add**.
   - **2c** Click the **Select trusted root(s)** icon, select the public certificate of the CA that signed the Identity Server certificates, then click **OK**.
   - **2d** Specify an alias, then click **OK** twice.
3. To apply the changes, click **Close**, then click **Update** on the Access Manager Appliance page.

Prerequisites for SSL Communication between the Access Gateway and Web Servers

If you are going to set up SSL between the Access Gateway and the Web servers, you need to configure your Web servers for SSL. Your Web servers must supply a certificate that clients (in this case, the Access Gateway) can import. See your Web server documentation for information about how to configure the Web server for SSL.

For mutual SSL, the proxy service must supply a certificate that the Web server can trust. This certificate can be the same one you use for SSL between the browsers and the reverse proxy.

Configuring SSL Communication with Browsers and the Identity Server

1. In Administration Console Dashboard, click **Devices** > **Identity Servers** > **Edit**.
2. Change **Protocol** to HTTPS (the system changes the port to 8443).
3. In the **SSL Certificate** line, click the **Browse** icon > **Replace** and select the Identity Server certificate.
4. Restart Tomcat.

   If your Identity Server and Administration Console are on the same machine, log in to Administration Console again.
5. After the Identity Server health turns green, go to **Access Gateway > Edit > Service Provider Certificates > Trusted Roots**.
6. Click **Add** to select the trusted root certificate of the certificate authority that signed Identity Server certificate.

   (Conditional) If you imported intermediate certificates for the CA, select them also.

   IMPORTANT: If the external certificate authority writes the DN in reverse order (the cn element is displayed first), you receive an error message that the certificate names do not match. You can ignore this warning, if the order of the DN elements is the cause.

   ---
7. Update Access Gateway.

Configuring SSL between the Proxy Service and the Web Servers

SSL must be enabled between the Access Gateway and browsers before you can enable it between the Access Gateway and its Web servers.

1. In the Administration Console, click **Devices** > **Access Gateways** > **Edit** > **[Name of Reverse Proxy]** > **[Name of Proxy Service]** > **Web Servers**.
2. To configure SSL, select **Connect Using SSL**.

   This option is not available if you have not set up SSL between browsers and the Access Gateway. See “Configuring SSL Communication with Browsers and the Identity Server” on page 828 and select the **Enable SSL between Browser and Access Gateway** field.

3 (Optional) Set up mutual authentication so that the Web server can verify the proxy service certificate:

3a Click the **Select Certificate** icon,

3b Select the certificate you created for the reverse proxy, then click **OK**.

This is only part of the process. You need to import the trusted root certificate of the CA that signed the proxy service's certificate to the Web servers assigned to this proxy service. For instructions, see your Web server documentation.

4 In the **Connect Port** field, specify the port that your Web server uses for SSL communication. The following table lists some common servers and their default ports.

Server Type	Non-Secure Port	Secure Port
Web server with HTML content	80	443
WebSphere	9080	9443
JBoss	8080	8443

5 To save your changes to browser cache, click **OK**.

6 To apply your changes, click the **Access Gateways** link, then click **Update** > **OK**.

Configuring the SSL Communication

By default, Access Manager Appliance supports the 128-bit SSL communication among the Administration Console, Identity Server, and browsers. The supported ciphers include:

- SSL_RSA_WITH_RC4_128_MD5
- TLS_DHE_DSS_WITH_AES_128_CBC_SHA
- SSL_RSA_WITH_3DES_EDE_CBC_SHA
- SSL_DHE_RSA_WITH_3DES_EDE_CBC_SHA
- SSL_DHE_DSS_WITH_3DES_EDE_CBC_SHA
- TLS_KRB5_WITH_3DES_EDE_CBC_SHA
- TLS_KRB5_WITH_RC4_128_SHA
- TLS_DHE_RSA_WITH_AES_128_CBC_SHA
- SSL_RSA_WITH_RC4_128_SHA
- TLS_RSA_WITH_AES_128_CBC_SHA

To enable weak ciphers (not recommended):

1 Modify the `server.xml` file located in `/opt/novell/nam/adminconsole/conf/`.

2 Add name of the ciphers that you want to enable in the ciphers tag.

To enable the strong 256-bit ciphers:

1 Download the Java Cryptography Extension (JCE) Unlimited Strength Jurisdiction Policy Files 7 from the Oracle website.

2 Extract the zip file and replace the policy jars in `/opt/novell/java/jre/lib/security/`.

3 Modify the `server.xml` file located in `/opt/novell/nam/adminconsole/conf/`.

4 Add the 256-bit ciphers to the cipher attribute of `<Connectors>`.

For example,

```
<Connector NIDP_Name="connector" port="2443" maxHttpHeaderSize="8192"
maxThreads="200" minSpareThreads="5" enableLookups="false"
disableUploadTimeout="true" acceptCount="0" scheme="https" secure="true"
clientAuth="false" sslProtocol="tls" URIEncoding="UTF-8"
allowUnsafeLegacyRenegotiation="false" keystoreFile="/var/opt/novell/novlwww/
.keystore" keystorePass="changeit" SSLEnabled="true" address="164.99.87.129"
ciphers="SSL_RSA_WITH_RC4_128_MD5, SSL_RSA_WITH_RC4_128_SHA,
TLS_RSA_WITH_AES_128_CBC_SHA, TLS_DHE_RSA_WITH_AES_128_CBC_SHA,
TLS_DHE_DSS_WITH_AES_128_CBC_SHA, SSL_RSA_WITH_3DES_EDE_CBC_SHA,
SSL_DHE_RSA_WITH_3DES_EDE_CBC_SHA, SSL_DHE_DSS_WITH_3DES_EDE_CBC_SHA" />
```

For a complete list of supported cipher suites and their requirements, see “The SunJSSE Provider”.

Maintaining Access Manager

Topics include:

- Chapter 16, “Auditing,” on page 833
- Chapter 17, “Reporting,” on page 843
- Chapter 18, “Logging,” on page 847
- Chapter 19, “Monitoring Component Statistics,” on page 873
- Chapter 20, “Monitoring Server Health,” on page 901
- Chapter 21, “Monitoring Component Command Status,” on page 907
- Chapter 22, “Monitoring Alerts,” on page 911
- Chapter 23, “Monitoring Access Manager By Using Simple Network Management Protocol,” on page 917
- Chapter 24, “Back Up and Restore,” on page 925
- Chapter 25, “Code Promotion,” on page 931
- Chapter 26, “Troubleshooting,” on page 943

16 Auditing

NOTE: Platform Agent and Novell Audit are no longer supported. A new Access Manager 4.2 installation no longer installs Platform Agent and Novell Audit for auditing. If you upgrade from an older version of Access Manager to 4.2, Platform Agent is still available. It is recommended to use Syslog for auditing.

Access Manager Appliance supports audit logging and file logging at the component level.Access Manager Appliance provides compliance assurance logging and to maintain audit log entries that can be subsequently included in reports. In addition to selectable events, device-generated alerts are automatically sent to the audit server. You can configure Access Manager Appliance to use a Sentinel server, a Sentinel Log Manager server or a third party syslog server.

The audit logs record events that have occurred in the identity and access management system and are primarily intended for auditing and compliance purposes. You can configure the following types of events for logging:

- Starting, stopping, and configuring a component
- Success or failure of user authentication
- Role assignment
- Allowed or denied access to a protected resource
- Error events
- Denial of service attacks
- Security violations and other events necessary for verifying the correct and expected operation of the identity and access management system

Audit logging does not track the operational processing of the Access Manager Appliance components; that is, the processing and interactions between Access Manager Appliance components required to fulfill a user request. (For this type of logging, see “Configuring Logging for Identity Server” on page 852.) Audit logs record the results of user and administrator requests and other system events. Although the primary purpose for audit logging is for auditing and compliance, you can also use the event logs for detecting abnormal and error conditions. The event logs can be used as a first alert mechanism for system support.

Access Manager Appliance has been assigned the server-alert event code 0x002E0605. It is responsible for packaging and forwarding audit log entries to the configured audit server.

For a secure system, you need to configure auditing and syslog to notify the system administrator when certain events occur. The most important audit events to monitor are the following:

- Configuration changes
- System shutdowns and startups
- Server imports and deletes
- Intruder lockout detection (available only for eDirectory user stores)
- User account provisioning

Audit events are device-specific. You can select events for the following devices:

- **Administration Console:** In the Administration Console, click **Auditing**.

- **Identity Server:** In the Administration Console, click **Devices** > **Identity Servers** > **Edit** > **Logging.**
- **Access Gateway:** In the Administration Console, click **Devices** > **Access Gateways** > **Edit** > **Auditing.**

This section discusses the following topics:

- "Enabling Auditing" on page 834
- "Enabling Identity Server Audit Events" on page 837
- "Enabling Access Gateway Audit Events" on page 840

Enabling Auditing

You can configure Access Manager Appliance to send audit events to a Sentinel server, a Syslog server, Novell Audit Server (on upgraded systems only), or a Sentinel Log Manager.

In addition to the selectable events, device-generated alerts are automatically sent to the audit server. These Management Communication Channel events have an ID of 002e0605. All Access Manager events begin with 002e.

For information about audit event IDs and field data, see "Access Manager Audit Events and Data" on page 1045.

The Access Gateway also supports sending e-mail notification to system administrators. To configure this system in the Administration Console, click **Devices** > **Access Gateways** > **Edit** > **Alerts.**

NOTE: The eDirectory audit configuration remains unchanged even after you upgrade to the latest version of the Access Manager. To fetch eDirectory audit events, you have to manually unload and re-load the audit modules. Perform this activity each time you start the eDirectory.

To install and enable eDirectory packages, see Installing Novell Audit Packages in the NetIQ eDirectory 8.8 SP8 Administration Guide.

Configuring Access Manager Appliance for Auditing

By default, Access Manager Appliance is preconfigured to use syslog. If you install more than one instance of for failover, syslog server is installed with each instance. However, if you already use a third party syslog server, you can configure Access Manager Appliance to use your audit server.

Access Manager Appliance allows you to specify only one audit server. You still have failover if the audit server is not reachable. The failover mechanism changes based on the type of logging:

- For File-based: Does not require a failover mechanism.
- For Syslog: The events are sent to a local file. The syslog client must be configured for failover. For more information, see the third party syslog server documentation.

If you have a Sentinel server or a Sentinel Log Manager server, you can configure Access Manager Appliance to send the events to them.

This section includes the following topics:

- "Specifying the Logging Server and Console Events" on page 835
- "Configuring Syslog for Auditing" on page 836

Specifying the Logging Server and Console Events

The Secure Logging Server manages the flow of information to and from the auditing system. It receives incoming events and requests, logs information to the data store, monitors designated events, and provides filtering and notification services. It can also be configured to automatically reset critical system attributes according to a specified policy.

1 To specify the logging server, click **Auditing.**

2 Fill in the following fields:

Audit Messages Using: Select any one of the following:

- **Platform Agent (Discontinued):** (Only on an Upgraded Access Manager)
- **Log File (Not Recommended For Production):** (Only on a fresh installation of Access Manager)

  The audit events are sent to a local log file. The file locations are:

  On Windows:

  - Identity Server and ESP: `"C:\Program Files(x86)\Novell\Syslog\audit_common.log"`
  - Access Gateway: `"C:\Program Files\Novell\Syslog\audit_ag.log"`

  On Linux:

  - Identity Server and ESP: `/var/opt/novell/syslog/audit_common.log`
  - Access Gateway: `/var/opt/novell/syslog/audit_ag.log`
- **Syslog**: From the list, select a syslog server. The available options are:
  - **Send to Sentinel**: The audit event are sent in CSV format.
  - **Send to Third party**: The audit events are sent in JSON format.

    NOTE: If the Administration Console is configured as a remote Audit server for syslog, then, the audit logs are sent to the following location: `/var/log/NAM_Audits.log`.

Server Listening Address: Specify the IP address or DNS name of the audit logging server you want to use. By default, the system uses the primary Administration Console IP address. If you want to use a different Secure Logging Server, specify that server here. For example, if you select syslog, specify the syslog server details here.

NOTE

- Access Manager supports auditing through syslog only on TCP.
- On Windows, if syslog is selected for auditing, the Server Listening Address field is disabled. To specify the server details, manually install and configure the local syslog client.

Server Public NAT Address: If your auditing server is in the private network, then you have to enter Public NAT IP Address of the auditing server using which devices can reach the auditing server.

To use a Sentinel server or a Sentinel Log Manager, specify the IP address or DNS name of the Sentinel.

- For more information about Sentinel, see Sentinel 7.3.
- For more information about Sentinel Log Manager, see Sentinel Log Manager 1.0.

Port: Specify the port that the Platform Agents or syslog uses to connect to the Secure Logging Server.

- For Platform Agent: The default secure logging server port is 1289.
- For Syslog:
  - For Sentinel server, the default port is 1468.
  - For third party syslog servers, specify the configured port of that server.

NOTE: If you select Sentinel server for auditing through syslog, then you must use the latest NetIQ Access Manager Collector for Sentinel.

Stop Service on Audit Server Failure: Enable this option to stop the Apache services when the audit server is offline or not reachable and audit events could not be cached.

3 Under **Management Console Audit Events**, specify the system-wide events you want to audit:

Select All: Selects all of the audit events.

Health Changes: Generated whenever the health of a server changes.

Server Imports: Generated whenever a server is imported into the Administration Console.

Server Deletes: Generated whenever a server is deleted from the Administration Console.

Configuration Changes: Generated whenever you change a server configuration.

4 Click **OK**.

If you did not change the address or port of the Secure Logging Server, this completes the process. It might take up to fifteen minutes for the events you selected to start appearing in the audit files.

Configuring Syslog for Auditing

NetIQ Access Manager supports syslog for auditing. You can use a Sentinel server or a third party syslog server to send audit events. To configure syslog, see “Specifying the Logging Server and Console Events” on page 835.

On Linux, if you select syslog for auditing, then the syslog server configurations are automatically pushed to Identity Server and Access Gateway.

On Windows, you need to manually install the preferred syslog service and configure it to communicate to the local TCP port 1290. To configure the syslog agent to communicate with the remote syslog server, you need to manually configure the installed syslog agent on each device.

Prerequisites

To view the prerequisites, see Upgrading Access Manager Appliance.

NOTE: To use Syslog for auditing, you need to upgrade the base operating system. After the upgrade, install the Syslog RPMs manually. To install the RPMs, execute the following command: `zypper in -t pattern NetIQ-Access-Manager`.

Limitations

The following are the limitations of syslog:

- On Identity Server and ESP, the events are cached to a local file during a local audit failure. The file locations:

  On Windows:

  - Identity Server and ESP: `"C:\Program Files(x86)\Novell\Syslog\audit_common.log"`

  On Linux:

  - Identity Server and ESP: `/var/opt/novell/syslog/audit_common.log`
- The log forwarding of cached logs in not supported for Identity Server and ESP events.
- The failover mechanism communication does not work in the Access Gateway.

IMPORTANT:

- By default, syslog agents do not cache or queue the audit events when the remote syslog audit server is not reachable. You need to manually configure the local syslog agent for caching the events.
- By default, syslog agents are configured without SSL communication with the remote audit server. You need to manually configure SSL between local syslog agent and remote syslog audit server.

For more information about syslog configuration, see Syslog Configuration Whitepaper.

Enabling Identity Server Audit Events

The following steps assume that you have already set up auditing on your network. For more information, see “Configuring Access Manager Appliance for Auditing” on page 834.

1. In the Administration Console, click **Devices > Identity Server > Servers** > **Edit** > **Logging**.
2. In the **Novell Audit Logging** section, select **Enabled**.
3. Select the events for notification.

   Select All: Select this option for all events. Otherwise, select one or more of the following:

Event	Description
Login Provided	Generated when an identity provider sends authentication to a service provider. Role assignment audit events are included in authentication audit events for the Identity Server.
Login Provided Failure	Generated when an identity provider attempts to send authentication to a service provider but fails.
Login Consumed	Generated when a user is authenticated either locally or by an external identity provider. Role assignment audit events are included in authentication audit events for the Identity Server.
Login Consumed Failure	Generated when the Identity Server initiates authentication, but the process fails.

Event	Description
Logout Provided	Generated when an identity provider sends a logout request to a service provider that it has authenticated.
Logout Local	Generated when the Identity Server receives a logout command from the user.
Federation Request Sent	Generated when a service provider attempts to federate with an identity provider.
Federation Request Handled	Generated by the Identity Server when processing a request for federation.
Defederation Request Sent	Generated by the identity provider when a request for defederation is sent to another provider.
Defederation Request Handled	Generated when the Identity Server processes a request for defederation.
Register Name Request Handled	Generated when the Identity Server processes a request for changing a name identifier.
Attribute Query Request Handled	Generated by the Identity Server when processing an attribute request from a service provider.
Web Service Query Handled	Generated when a Web service query request is sent to an identity provider.
Web Service Modify Handled	Generated when Web service modify request is sent to an identity provider.
User Account Provisioned	Generated by the Identity Server when functioning as an identity consumer and when an account has been provisioned.
User Account Provisioned Failure	Generated by the Identity Server when functioning as an identity consumer and when account provisioning has failed.
LDAP Connection Lost	Generated when the LDAP connection is lost.
LDAP Connection Reestablished	Generated when the LDAP connection is reestablished.
Server Started	Generated when the server gets a start command from the server communications module.
Server Stopped	Generated when the server gets a stop command from the server communications module.
Server Refreshed	Generated when the server gets a refresh command from the server communications module.
Intruder Lockout Detected	Generated when an attempt to log in as a particular user with an invalid password has occurred more times than is allowed by the directory.
Component Log Severe Messages	Logged for all component messages with level of Severe.
Component Log Warning Messages	Logged for all component messages with level of Warning.
Brokering Across Groups Denied	Brokering authentication request denied to a target service provider. The brokering group consists of either the Identity Provider or target Service Provider, but both does not belong to the same group.

Event	Description
Brokering Rule Evaluated to Deny	Brokering authentication request denied to a target service provider due to broker policy evaluation resulted in denying.
Brokering Handled	The total number of brokering authentication requests handled by the Identity Server when it started.
WebService Request Authenticated	Generated when a user is authenticated for requesting a token for a Web service.
WebService Request Authentication Failed	Generated when a user's authentication fails for requesting a token for a Web service.
Token Was Issued To WebService	Generated when a token is issued for accessing a Web service.
Token Issue To WebService Failed	Generated when a request to issue a token for accessing a Web service fails.
Token Was Validated To A WebService	Generated when a token is validated for a Web service.
Token Validation To WebService Failed	Generated when a token validation for accessing a web service fails.
Token Renewed	Generated when a token is renewed for a Web service.
Token Renew Failed	Generated when renewing a token for a Web service fails.
Oauth & OpenID Token Issued	Generated when an OAuth Authorization code, OAuth token, ID token, or Refresh token is issued.
Oauth & OpenID Token Issue Failed	Generated when OAuth Authorization code issue, OAuth token issue, ID Token issue, or Refresh token issue failed.
Oauth Consent Provided	Generated when OAuth consent is provided to a client application.
Oauth Consent Revoked	Generated when OAuth consent is revoked from a client application.
OAuth Client Applications	Generated when a client is registered, updated, deleted, or client registration fails.
Oauth & OpenID Token Validation Success	Generated when an OAuth and OpenID token is validated successfully.
Oauth & OpenID Token Validation Failed	Generated when an OAuth and OpenID token validation fails.
Risk-Based Authentication Succeeded	Generated when rule execution succeeds.
Risk-Based Authentication Failed	Generated when rule execution fails.
Risk-Based Authentication Action Involved	Generated when rule execution succeeds and user is requested to perform additional authentication.

4 Click **Apply** > **OK**.

5 Click **Servers** > **Update Servers**.

NOTE: Identity Server logs the IP address of client machine, from where authentication requests originate, into audit events. If the client machine is behind a proxy, then proxy IP address will be logged. If you need to skip the proxy IP address and log the actual client machine IP address, you have to configure the RemoteIpValve in the Tomcat configuration file (`server.xml`) on all the identity servers.

The `server.xml` file can be found at `/opt/novell/nam/idp/conf/server.xml` (for linux) and `// Program File x(86)/Novell/Tomcat/conf/server.xml` (for Windows).

The details of configuring RemoteIpValve can be found at Remote IP Valve (http://tomcat.apache.org/tomcat-8.0-doc/config/valve.html#Remote_IP_Valve).

To configure audit events to capture the source IP address of the X-forwarded-header, add the following details after the Engine element in the `server.xml` file:

```
<Engine defaultHost="localhost" name="Catalina">

<Valve className="org.apache.catalina.valves.RemoteIpValve"

internalProxies="IP addresses" />
```

Substitute IP addresses with the IP address of the proxy and load balancer.

Restart Tomcat by using `rcnovell-idp` restart.

Enabling Access Gateway Audit Events

The following steps assume that you have already set up auditing on your network. For more information, see "Configuring Access Manager Appliance for Auditing" on page 834.

1. In the Administration Console, click **Devices** > **Access Gateways** > **Edit** > **Auditing**.
2. Based on the event category, select the events for notification.

   Select All: Select this option for all events. Otherwise, select one or more of the following:

   Events

Event	Description
Access Denied	Generated when a requested action is denied because the requester has insufficient access rights to a URL.
Identity Injection Failed	Generated when the Identity Injection policy injects with the value field empty.
System Started	Generated when the Access Gateway is started.
System Shutdown	Generated when the Access Gateway is stopped.
Form Fill Failed	Generated when a Form Fill policy fails to successfully fill in a form.
Application Accessed	Generated when a user accesses applications.
URL Not Found	Generated when a requested URL cannot be found.
IP Access Attempted	Generated when a user attempts to access a URL with an IP address instead of the published DNS name configured in the Access Gateway.

Event	Description
Oauth & OpenID Token Validation Failed	Generated when an OAuth and OpenID token validation fails.

High Volume Events

Event	Description
Access Allowed	Generated when a requested action is allowed because the requester has the correct access rights to a URL.
Identity Injection Success	Generated when the Identity Injection policy successfully injects data into the HTTP header.
Form Fill Success	Generated when a Form Fill policy successfully fills in a form.
URL Accessed	Generated when a user accesses a URL.

IMPORTANT: Enabling **High Volume Events** may generate high volume of audit events on a system with heavy load. This may degrade the performance of the Access Gateway.

Audit Filters

Select the following filters to exclude them from the audit events:

Filter	Description
CSS	Excludes CSS files as part of response from the audit events.
JavaScripts	Excludes JavaScripts from the audit events.
Images	Excludes images from the audit events. Specify the image format. For example: JPEG, PNG
URLs Matching Regular Expression	Excludes URLs matching the configured regular expression. It helps in filtering out specified URL paths from the ones audited as part of the URL Accessed audit event. These filtered out URL paths will not be displayed in the Audit Server. This is helpful where auditing every URL is not required and may increase the load on the Audit Server. The regular expression is standard perl based regular expressions. For more information about “Regular Expressions”, see http://perldoc.perl.org/perlre.html. Each URL (path?querystring) is matched against this expression. If the match is successful, the URL will not be audited for URL access. For example: ◆ To exclude the health check messages from auditing: `/nesp/app/heartbeat` ◆ To exclude the auditing of URL under the path /images/: `/images/*`

3 To save your modifications, click **OK** twice.

4 On the Access Gateways page, click **Update**.

17 Reporting

- “Overview” on page 843
- “Prerequisites” on page 844
- “Deploying Access Manager Reporting Solution Pack” on page 845
- “Enabling Reporting” on page 845
- “Generating Reports” on page 846

Overview

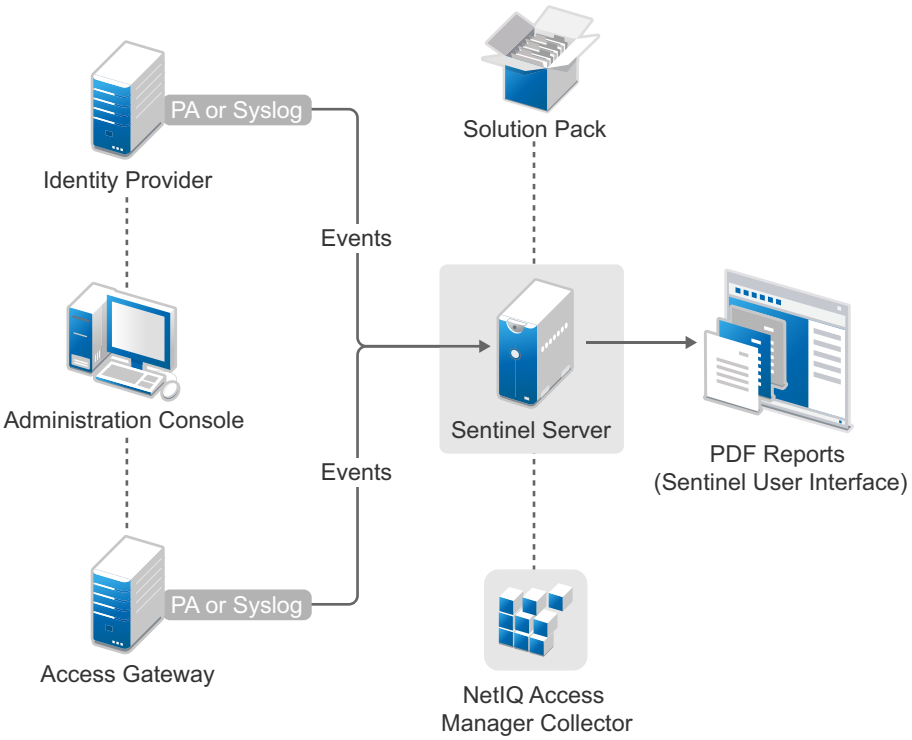
Access Manager Appliance uses Sentinel Solution Pack to generate reports. This solution pack consists of predefined report definitions. Access Manager Appliance requires NetIQ Sentinel or Sentinel Log Manager to use this feature. You can use these reports to analyze users’ accesses to applications protected by Access Manager, in auditing, and for compliance purposes.

NOTE: Platform Agent and Novell Audit are no longer supported. A new Access Manager 4.2 installation no longer installs Platform Agent and Novell Audit for auditing. If you upgrade from an older version of Access Manager to 4.2, Platform Agent is still available. It is recommended to use Syslog for auditing.

You can generate and download the following canned reports:

- NetIQ Access Manager Application Access Summary
- NetIQ Access Manager Application Specific User Access
- NetIQ Access Manager Federation Summary
- NetIQ Access Manager User Application Access Summary
- NetIQ Access Manager User Login Contract Summary
- NetIQ Access Manager User Login Failure Report
- NetIQ Access Manager Application Specific Risk based Authentication Report

The following diagram illustrates the Access Manager Appliance reporting architecture when integrated with Sentinel:



- When an event occurs in the Identity Server or Access Gateway, Platform Agent (PA) or Syslog sends it to Sentinel.
- In Sentinel, NetIQ Access Manager Collector parses these events and saves event data in the Sentinel database.
- The NetIQ Access Manager Reporting Solution Pack provides predefined report templates. You can use this template against event data to generate PDF reports.
- You can access a report through the Sentinel user interface by using a web browser.

Prerequisites

The following list includes prerequisites for using Access Manager Reporting Solution Pack:

- Install Sentinel 7.1 (or later) or Sentinel Log Manager 1.2.2 (or later).

  For information about how to install Sentinel, see the Sentinel Installation Guide.

  For information about how to install Sentinel Log Manager, see the Sentinel Log Manager Installation Guide.
- Deploy Access Manager Reporting Solution Pack. See “Deploying Access Manager Reporting Solution Pack” on page 845.
- Deploy NetIQ Access Manager Collector Pack for Sentinel. The collector pack is available at the NetIQ Sentinel Plug-ins site.

Deploying Access Manager Reporting Solution Pack

To use the predefined report templates of the solution pack, you must deploy Access Manager Reporting Solution Pack in the Sentinel system. The solution pack is available at the NetIQ Sentinel Plug-ins site.

For information about how to deploy Access Manager Reporting Solution Pack, see the Access Manager Reporting Solution Pack Guide.

After deploying the solution pack, perform the following steps to verify that Access Manager reports are added:

1. Log in to Sentinel or Sentinel Log Manager.
2. Click **Reports and Searches**.
3. Verify that NetIQ Access Manager reports are listed.

Enabling Reporting

To enable the reporting feature, perform the following steps:

1 **Enable Events:** Enable Application Access, Federate Request Sent, Federation Request Handled, Login Consumed, Login Consumed Failed, Risk-based Authentication Succeeded, Risk-based Authentication Failed, and Risk-based Authentication Action Invoked events.

For information about how to enable the Application Access event, see “Enabling Access Gateway Audit Events” on page 840.

For information about how to enable Federation Request Sent, Federation Request Handled, Login Consumed, Login Consumed Failure, Risk-based Authentication Succeeded, Risk-based Authentication Failed, and Risk-based Authentication Action Invoked events, see “Enabling Identity Server Audit Events” on page 837.

The following table lists Access Manager reports and associated events:

Name of Report	Description	Event	Component
NetIQ Access Manager Application Access Summary	Summary of applications accessed at a specified time	Application Access	Access Gateway
NetIQ Access Manager User Application Access Summary	Users who accessed a particular application at a specified time	Application Access	Access Gateway
NetIQ Access Manager Application Specific User Access	Number of applications accessed by a specific user at a specified time	Application Access	Access Gateway
NetIQ Access Manager Federation Summary	Users who accessed a federated service at a specified time	Federation Request Sent and Federation Request Handled	Identity Server
NetIQ Access Manager User Login Contract Summary	Number of user login based on authentication contracts at a specified time	Login Consumed	Identity Server

Name of Report	Description	Event	Component
NetIQ Access Manager User Login Failure Report	Number of failed login attempts and their reasons	Login Consumed Failure	Identity Server
NetIQ Access Manager Application Specific Risk based Authentication Report	Number of risk-based authentication attempts and the action taken for each attempt at a specified time for a specific application.	Risk-based Authentication Succeeded Risk-based Authentication Failed Risk-based Authentication Action Invoked	Identity Server

2 **Configure the IP Address of Sentinel Server or Sentinel Log Manager in Access Manager:** Perform the following steps:

2a Log in to Access Manager.

2b Click **Auditing**.

2c Specify the following details:

Server Listening Address: Specify the Sentinel server IP address.

Port: Specify the default port of NetIQ Access Manager Collector.

2d Click **Apply** > **OK**.

Generating Reports

In Sentinel, after deploying NetIQ Access Manager Solution Pack and Collector, you can generate a report from available pre-defined report templates and search for events based on the report definitions. You can also schedule, export, and email the reports.

For information about how to generate, schedule, search, export, manage, and delete a report, see Reporting in the NetIQ Sentinel User Guide or Reporting in the Sentinel Log Manager Administration Guide depending on the tool you are using.

NOTE: For sample reports, see Section D, “Access Manager Reports Samples,” on page 1197.

18 Logging

Logging is the main tool you use for debugging the Access Manager configuration. You can enable and configure how the system performs logging. All administrative and end-user actions and events are logged to a central event log. This allows easy access to this information for security and operational purposes. Additionally, the log system provides the ability to monitor ongoing activities such as identity provider authentication activity, up-time of the system, and so on. File logging is not enabled by default.

Each Access Manager Appliance device has configuration options for logging:

Identity Server: Logging is turned off and must be enabled. When you enable Identity Server logging, you also enable logging for the Embedded Service Providers that are configured to use the Identity Server for authentication. For configuration information, see “Configuring Logging for Identity Server” on page 852.

Embedded Service Providers: Each Access Manager Appliance device has an Embedded Service Provider that communicates with the Identity Server. Its log level is controlled by configuring Identity Server logging.

Access Gateway Service: The Gateway Service logs contain the messages sent between the Gateway Service and the Embedded Service Provider and between the Gateway Service and the Web server. This type of logging is turned off and must be enabled. For information, see “Managing Access Gateway Logs” on page 860.

This sections discusses the following topics:

- “Understanding the Types of Logging” on page 847
- “Understanding the Log Format” on page 849
- “Identity Server Logging” on page 852
- “Access Gateway Logging” on page 860
- “Downloading Log Files” on page 869
- “Turning on Logging for Policy Evaluation” on page 871

Understanding the Types of Logging

Access Manager Appliance supports two types of logging:

- “Component Logging for Troubleshooting Configuration or Network Problems” on page 847
- “HTTP Transaction Logging for Proxy Services” on page 848

Component Logging for Troubleshooting Configuration or Network Problems

Each Access Manager Appliance component maintains log files that contain entries documenting the operation of the component. Component file logging records the processing and interactions between the Access Manager components that occur while satisfying user and administrative requests and during general system processing. By enabling the correct levels of logging for the various Access

Manager components, an administrator can monitor how the Access Manager Appliance processes user and administrative requests. Transaction flows have been defined to help the administrator identify the processing steps that occur during the execution of specific types of user or administrative requests. All component file logs include tags and values that allow the administrator to identify and correlate which component file log entries pertain to a given transaction and user.

Component file logs are not primarily intended for debugging the software itself, although they can be used to detect software that is not behaving properly. Rather, the intent of component file logging is to document the operational processing of the Access Manager components so that system administrators and support personnel can identify and isolate problems caused by configuration errors, invalid user data, or network problems such as broken connections. However, component file logging is typically the first step in identifying software bugs.

Component file logging is more verbose than audit logging. It increases processing load, and on a day-to-day basis, it should be enabled only to log error conditions and system warnings. If a specific problem occurs, component file logging can be set to **info** or **config** to gather the information needed to isolate and repair the detected problem. When the problem is resolved, component file logging should be reconfigured to log only error conditions and system warnings.

Log files can be configured to include entries for the following events:

- Initialization and shutdown
- Configuration
- Events processed by the component, such as authentication, role assignment, resource access, and policy evaluation
- Error conditions

See “Configuring Logging for Identity Server” on page 852.

HTTP Transaction Logging for Proxy Services

The Access Gateway allows you to log HTTP transactions. You can log what happens with an HTTP request and response during certain times:

- Between the browser and the Access Gateway
- Between the Access Gateway and the back-end Web server

You select fields from the HTTP header of a request and these fields are logged. You can then use these logged transactions to bill customers for Web services or to troubleshoot whether a request is refused because the browser did not ’send the required information or because the Access Gateway did not’ send the Web server the required information.

This type of logging conforms to the W3C specification for proxy server logging in the common and extended log formats. This type of logging provides no information about the exchanges between the Access Gateway and the Identity Server. If you need to discover whether the Access Gateway is obtaining the correct information from the Identity Server for an Identity Injection or Form Fill policy, you need to turn on component logging. See “Configuring Logging for Identity Server” on page 852.

For HTTP transaction logging, see “Configuring Logging for a Proxy Service” on page 861.

Understanding the Log Format

Access Manager Appliance does not have a fixed format for file log entries. However, to facilitate the use of non-interactive stream-oriented editors such as sgrep, sed, awk, and grep and to improve log entry readability, the log entries in the `catalina.out` files use some standard elements. These entries use the beginning and ending log entry tags and the log entry correlation tags. The data portion of log entries is the most flexible part. A log entry has the following fields:

```
<amLogEntry> [\n]
   time-date-stamp
   [log preamble]:
   AM#event-code:
   AMDEVICE#device-id:
   AMAUTHID#auth-id:
   AMEVENTID#event-id:
   [..additional correlating information][\n]
   [supplementary log entry data and text ... \n]
</amLogEntry> [\n]
```

Most log entries do not use the optional line breaks (`[\n]`). Notice that the time-date-stamp, the log preamble, the correlation tags, and optional additional correlating information are on the same line so that stream-oriented editors that use only one line (such as grep) can be used to locate log entries that are related. The following entry is an example entry that is logged when a user has initiated a login sequence.

```
<amLogEntry> 2009-06-08T21:06:25Z INFO NIDS Application: AM#500105014:
AMDEVICEID#9921459858EAAC29: AMAUTHID#BB11C254B7521B5E836D8703826287 AF:
Attempting to authenticate user cn=jwilson,o=novell with provided credentials. </
amLogEntry>
```

Table 18-1 *Fields in a Log Entry*

Field	Description
Beginning, ending tags	The `<amLogEntry>` and `</amLogEntry>` tags mark the beginning and the end of a log entry. These tags allow stream-oriented editors to extract log entries for processing.
Time-date-stamp tag	The date and time is specified in the W3C profile format of ISO 8061. It has the following fields: year-month-day-T-hour-minutes-seconds-time zone. The Z value for the time zone indicates that the time is specified in UTC.
Log preamble	This information is optional, and usually consists of a string indicating the logging level (such as warning, informational, or debug) and a string identifying the type of module making the entry. In the example log entry, the preamble has a log level and a module identifier and contains the following strings: `INFO NIDS Application`:
Correlation tags	The correlation tags uniquely identify the event, the device that produced the event, and the user who requested the action. The example log entry contains the following correlation tags: AM#500105014: AMDEVICEID#9921459858EAAC29: AMAUTHID#BB11C254B7521B5E836D8703826287AF: For more information, see “Understanding the Correlation Tags in the Log Files” on page 850.

Field	Description
Additional correlation information	This information is optional and contains correlation tags and data unique to a specific type of trace. For example, a policy evaluation trace created by the Embedded Service Provider contains the following additional tags: ◆ NXPESID#value ◆ POLICYID#value The example log entry does not contain any additional correlation information. For a log entry that does, see “Identity Injection Traces” on page 1037.
Supplementary information	This information is optional and contains information that is specific to the log entry. It can be as simple as an informational string, such as the string in the example log entry: `Attempting to authenticate user cn=jwilson,o=novell with provided credentials.` The supplementary information can have a very specific format. For an example and explanation of the policy trace information, see “Understanding Policy Evaluation Traces” on page 1028.

Understanding the Correlation Tags in the Log Files

There is no fixed field format for log file entries. However, because most requests handled by Access Manager Appliance are processed by multiple Access Manager Appliance components, there is a mechanism that facilitates the correlation of log entries for a single Access Manager Appliance request in the various component log files. A correlation tag has the following general format:

`<tag name>#<tag value>:`

The `<tag name>` is a fixed value, defined in the Format column of Table 18-2. It is always terminated by the # character. The `<tag value>` immediately follows the # character and is always terminated by the : character. The `<tag value>` is not a fixed value, but a uniquely assigned value to identify an event, a user, or a transaction. Table 18-2 lists the defined correlation tags:

Table 18-2 *Correlation Tags*

Type	Format	Description
Event code	`AM#<Event-Code>:`	This tag is included in all log entries that record an event and in all events that are presented to the user as an informational or error page.

Type	Format	Description
User ID	`AMAUTHID#<ID>:`	An authentication identifier that the Identity Server or the Embedded Service Provider (ESP) assigns to each authenticated user. This tag is included in all entries that pertain to a request made by an authenticated user. Currently the Identity Server and ESP assign different authentication IDs. When correlating the flow of events between the Identity Server and the ESP for an authentication sequence, you can use the event code of the authentication events and find the artifact that the ESP and the Identity Server exchange. In the `catalina.out` file of the Identity Server, search for `AM#500105018` events. This is the event that sends the artifact to the ESP. Search for a corresponding artifact in the Access Gateway log. Events `AM#500105020` and `AM#500105021` contain the artifact value.
Device ID	`AMDEVICE#<ID>`	An identifier that uniquely identifies the Access Manager Appliance device that is generating the log entry. You can view the identifier that is assigned to each device on the General Logging page in the Administration Console (click **Auditing** > **General Logging**). The ID begins with a prefix that identifies the type of device such as idp for Identity Server, ag for an Access Gateway, and idp-esp for ESP of the device. The prefix is followed by a 16-digit hexadecimal number. In log entries, the idp prefix is not recorded. For example, the General Logging page displays `idp-AA257DA77ED48DB0` for the ID of the Identity Server, but in the `catalina.out` file, the value is `AMDEVICE#AA257DA77ED48DB0`.
Transaction ID	`AMEVENTID#<ID>:`	An identifier assigned to each Access Manager Appliance or system administration transaction. Access Manager Appliance transactions are actions such as authenticating a user, processing a request for access to a resource, and federating an identity. If a user requests access to multiple resources, each request is given a separate transaction ID. When the Access Gateway evaluates a policy for a protected resource page and the page contains links, the policy is evaluated for each link, and each of these evaluations generates a new transaction ID. System administration transactions are actions such as importing a device, deleting a device, stopping or starting a device, and configuring or modifying the configuration of a device.

Sample Scenario

The following scenario illustrates how these tags can be used. A user receives an error page indicating that the user has been refused access to a protected resource. The error page contains an event code. The user contacts the system administrator and reports the event code contained in the message. The code displayed to the user includes both an event number and an identifier indicating the device detecting the error, for example, `300101023-92E1B234`. The `300101023` value is the event number and `92E1B234` is the device identifier. The device identifier is the number assigned to the

Access Manager Appliance device reporting the error. You can make a textual search of log entries using the tags and values `AM#300101023:` and `AMDEVICEID#92E1B234:` to locate candidate log entries of the target Access Manager Appliance transaction flow. When the desired log entry is found, the `AMEVENTID#` tag and value and the `AMAUTHID# tag` (assuming the user has been authenticated) from the log entry can be used to locate all other log entries pertaining to the user in the context of the transaction.

Identity Server Logging

- "Configuring Logging for Identity Server" on page 852
- "Configuring Session-Based Logging" on page 854

Configuring Logging for Identity Server

If you change or enable logging, you must update the Identity Server configuration and restart the Embedded Service Providers to apply the changes. When you disable logging, you must also restart the Embedded Service Providers.

This section discusses the following topics:

- "Enabling Component Logging" on page 852
- "Managing Log File Size" on page 854

Enabling Component Logging

File logging records the actions that have occurred. For example, you can configure Identity Server logging to list every request made to the server. With log file analysis tools, you can get a good idea of where visitors are coming from, how often they return, and how they navigate through a site. The content logged to file logging can be controlled by specifying logger levels and by enabling statistics logging.

1. In the Administration Console, click **Devices > Identity Servers > Edit > Logging**.
2. **File Logging:** The following options are available for component logging:
   - **Enabled:** Enables file logging for this server and its associated Embedded Service Providers.
   - **Echo To Console:** Copies the Identity Server XML log file to `/var/opt/novell/nam/logs/idp/tomcat/catalina.out` (Linux). You can download the file from **Auditing > General Logging.**

     For the Embedded Service Providers, the log file location depends upon the device:
     - For an Access Gateway Appliance or a Linux Access Gateway Service, this sends the messages to the `catalina.out` file of the device.
   - **Log File Path:** Specifies the path that the system uses to save the Identity Server XML log file. The default path is `/var/opt/novell/nam/logs/idp/nidplogs`.

     If you change this path, you must ensure that the user associated with configuring the identity or service provider has administrative rights to the Tomcat application directory in the new path.
   - **Maximum Log Files:** Specifies the maximum number of Identity Server XML log files to leave on the machine. After this value is reached, the system deletes log files, beginning with the oldest file. You can specify **Unlimited**, or values of 1 through 200. 10 is the default value.

- **File Wrap:** Specifies the frequency (hour, day week, month) for the system to use when closing an XML log file and creating a new one. The system saves each file based on the time you specify and attaches the date and/or time to the filename.
- **GZip Wrapped Log Files:** Uses the GZip compression utility to compress logged files. The log files that are associated with the **GZip** option and the **Maximum Log Files** value are stored in the directory you specify in the **Log File Path** field.

3 **Component File Logger Levels:** Specify the logging sensitivity for the following protocols:

Application: Logs system-wide events, except events that belong to a specific subsystem.

Liberty: Logs events specific to the Liberty IDFF protocol and profiles.

SAML 1: Logs events specific to the SAML1 protocol and profiles.

SAML 2: Logs events specific to the SAML2 protocol and profiles.

WS Trust: Logs events specific to the WS-Trust protocol.

WS Federation: Logs events specific to the WS Federation protocol.

OAuth and OpenID Connect: Logs events specific to the OAuth and OpenID Connect protocols.

Web Service Provider: (Liberty) Logs events specific to fulfilling Web service requests from other Web service consumers.

Web Service Consumer: (Liberty) Logs all events specific to requesting Web services from a Web service provider.

Use the drop-down menu to categorize logging sensitivity. Higher logging levels also include the lower levels in the log.

- **Off:** Turns off component file logging for the selected item.
- **Severe:** Logs serious failures that can cause system processing to not proceed.
- **Warning:** Logs potential failures, but the impact on execution is minimal. Warnings indicate that you should be aware that this event is happening and might want to make a configuration change to avoid it.
- **Info:** Logs informational events. No execution or data impact occurred.
- **Verbose:** Logs static configuration information. The system logs any configuration errors under one of the primary three levels: Severe, Warning, and Info.
- **Debug:** Includes all of the preceding levels.

4 **Statistics Logging:** (Optional) Enable this option if you want the system to periodically send the system statistics, in string format, to the current file logger. Statistical data (such as counts, levels, and so on) are included in the file log.

4a In the **Statistics Logging** section, select **Enabled**.

4b In the **Log Interval** field, specify the time interval in seconds that statistics are logged.

5 **Novell Audit Logging:** For information about configuring Novell Audit Logging, see “Enabling Identity Server Audit Events” on page 837.

6 Click **OK**.

7 Update the Identity Server.

8 Restart the Embedded Service Providers on the devices, in order to apply the changes.

When you disable component logging, you need to update the Identity Servers and restart the Embedded Service Providers.

Managing Log File Size

On Linux, the logrotate daemon manages the log files located in the following directories:

```
/opt/novell/nam/logs
/opt/volera/roma/logs/
```

The logrotate daemon has been configured to scan the files in these directories once a day. It rolls them over when they have reached their maximum size and deletes the oldest version when the maximum number of copies have been created.

If you want to modify this behavior, see the following files in the `/etc/logrotate.d` directory:

```
novell-idp
novell-devman
```

For information about the parameters in these files, see the documentation for the logrotate daemon.

Configuring Session-Based Logging

The session-based logging feature allows the administrator to enable file logging for an individual user. In production environments, this has the following value:

- Debug logging can be turned on for an individual user rather than all users. The potential size of logged data usually prohibits an administrator from turning on debug logging for all users.
- All logged messages for this user are directed to a single file. Administrators do not need to sort through the various log files to follow the activity of the user.
- Isolating the problem and finding the cause is limited to the user who is experiencing the problem.
- Enabling session-based logging does not require a configuration change to the Identity Server, and thus does not require updating the Identity Server.

The following user scenario explains how this feature could be used in a production environment

1. A user notices a problem and calls the help desk.
2. The help desk operator questions the users and concludes that the problem is caused by either a NetIQ Identity Server or an Embedded Service Provider.
3. The operator has been granted the rights to create logging tickets, and uses the User Portal to create a logging ticket for the user.
4. The operator sends the logging ticket password and the URL to access the logging ticket class to the user.
5. The user clicks the URL and enters the logging ticket password.

   This marks the current session as “active for logging” and adds a small icon to the top right of the page, which makes the session logging feature visible to the user.
6. Using the same browser window, the user duplicates the problem behavior.
7. The operator can then access the data that was logged just for this user and analyze the cause of the behavior.

To enable session-based logging, the following tasks need to be completed:

- “Creating the Administrator Class, Method, and Contract” on page 855
- “Creating the Logging Session Class, Method, and Contract” on page 856

- “Enabling Basic Logging” on page 857
- “Responding to an Incident” on page 857

Creating the Administrator Class, Method, and Contract

The IDP Administrator class, method, and contract control who has the rights to create a logging ticket. You need to know the DNs of the operators who are going to be responding to the users who are experiencing problems.

1 In the Administration Console, click **Devices** > **Identity Servers** > **Edit** > **Local.**

2 To create the class:

2a Click **Classes**.

2b Click **New**, then specify the following values:

Display name: `IDP Administrator`

Java class: `Other`

Java class path: com.novell.nidp.authentication.local.IDPAdministratorClass

2c Click **Next**, then click **Finish**.

3 To create the method:

3a Click **Methods**.

3b Click **New**, then specify the following values:

Display name: `IDP Administrator Method`

Class: `IDP Administrator`

Identifies user: Deselect this option.

User Stores: Select the user stores that contain your operators, then move them to the list of User Stores.

3c In the **Properties** section, click **New**, then specify the following to create an IDP Administrator:

Property Name: `Administrator1`

The Property Name must begin with Administrator; append a value to this so that each property has a unique value.

Property Value: `cn=jdoe,o=users`

The Property Value must be the DN of an operator in the user stores you selected in Step 3b. Use LDAP typed comma notation for the DN.

3d Repeat Step 3c for each IDP Administrator you require.

You can return to this method to add or remove IDP Administrators, when responsibilities change.

3e Click **Finish**.

4 To create the contract:

4a Click **Contracts**.

4b Click **New**, then specify the following values:

Display name: `IDP Administrator Contract`

URI: `urn:novell:nidp:admin:contract`

Methods: Move the **IDP Administrator Method** to the Methods list.

Leave all other fields with their default values.

4c Click **Next**, then specify the following values for the authentication card:

ID: `IDPAdmin`

Text: `IDP Administrator`

Image: Select an image from the list, such as the IDP Administrator image that was created for this type of contract.

Show Card: Deselect this option.

4d Click **Finish**.

5 Continue with "Creating the Logging Session Class, Method, and Contract" on page 856.

Creating the Logging Session Class, Method, and Contract

1 In the Administration Console, click **Devices** > **Identity Servers** > **Edit** > **Local.**

2 To create the class:

2a Click **Classes**.

2b Click **New**, then specify the following values:

Display name: `Logging Session`

Java class: `Other`

Java class path: com.novell.nidp.authentication.local.LogTicketClass

2c Click **Next**, then click **Finish**.

3 To create the method:

3a Click **Methods**.

3b Click **New**, then specify the following values:

Display name: `Logging Session Method`

Class: `Logging Session`

Identifies user: Deselect this option.

User Stores: Select the user stores that contain the users that potentially can experience problems, then move them to the list of User Stores.

3c Click **Finish**.

4 To create the contract:

4a Click **Contracts**.

4b Click **New**, then specify the following values:

Display name: `Logging Session Contract`

URI: `urn:novell:nidp:loggingsession:contract`

Methods: Move the **Logging Session Method** to the **Methods** list.

Leave all other fields with their default values.

4c Click **Next**, then specify the following values for the authentication card:

ID: `LogSession`

Text: `Logging Session`

Image: Select an image from the list, for example the Session Logging image that was created for this type of contract.

Show Card: Deselect this option.

4d Click **Finish**.

5 Click **OK**, then update the Identity Server.

6 Continue with "Enabling Basic Logging" on page 857.

Enabling Basic Logging

For session-based logging to function, logging on the Identity Server must be enabled. However, you do not need to select what is logged. The Logging Ticket enables the appropriate components and levels when an incident occurs.

1 In the Administration Console, click **Devices** > **Identity Servers** > **Edit**.

2 Click **Logging**, then specify the following:

File Logging: Enable this option.

Echo To Console: Enable this option.

No other options need to be enabled. The **Component File Logger Levels** can be left in their default state of off.

3 Click **OK**, then update the Identity Server.

This completes the configuration. You now need to wait for a user to report a problem. For information about using this feature to respond to a problem, see "Responding to an Incident" on page 857.

Responding to an Incident

The following sections explain how to use the feature when a user reports a problem:

- "Creating a Logging Ticket" on page 857
- "Enabling a Logging Session" on page 858
- "Viewing the Log File" on page 859

Creating a Logging Ticket

These steps are performed by an IDP Administrator when a user reports a problem:

1 Log in to the Identity Server by using the credentials of an administrator.

If the base URL of the Identity Server is https://idp.amlab.net:8443/nidp, enter the following URL:

```
https://idp.amlab.net:8443/nidp/app
```

2 Change to the Logging Ticket page by specifying the following URL:

```
https://idp.amlab.net:8443/nidp/app/login?id=IDPAdmin
```

The *id* specified in the URL must match the ID you specified for the ID of the IDP Administrator Contract. See Step 4c of "Creating the Administrator Class, Method, and Contract" on page 855.

If you logged in with the credentials of an IDP Administrator, an **Administrator** tab appears.

3 To create a ticket for the user, click the **Administrator** tab.

3a Click **New**.

3b Specify the following:

Ticket: Specify a name for ticket.

You must share this name with the user who reported the problem.

Ticket Good For: Select a time limit for the ticket, from one minute through one year.

When selecting a time limit, consider the following:

- When a ticket expires, logging is automatically stopped. If you know that user is experiencing a problem that prevents the user from logging out, you might want to create a ticket with a short time limit.
- If the user does not log out (just closes the browser window or the problem closes it), the session remains in the list of logged sessions. After 10 minutes of inactivity, the session is closed and the lock on the log file is cleared. As long as the log file is locked, no other application can read the file.

Ticket Log Level: Select the level of information to log, from severe-only messages to debug.

Log to Console: Select to log the messages to the user's file and to the console.

- If you have set up logging for session-based logging (see "Enabling Basic Logging" on page 857), then this allows you see the messages in the `catalina.out` or `stdout.log` file.
- If you have enabled Component File Logger Levels, selecting this option can create duplicate entries in the `catalina.out` or `stdout.log` file.

3c Click **Create**.

4 Create a URL that uses the following format:

`https://<base_URL>/nidp/app/login?id=<LogSession>`

Replace <*base_URL*> with the base URL of your Identity Server, including the port. Ensure that the port agrees with the HTTP scheme (either http or https).

Replace <*LogSession*> with the ID you specified for the authentication card when defining the Logging Session contract.

IMPORTANT: The id is the ID of the authentication card of the Logging Session contract (see Step 4c of "Creating the Logging Session Class, Method, and Contract" on page 856). It is not the name of the ticket you just created.

If the base URL of the Identity Server is https://idp.amlab.net:8443/nidp and the ID for the authentication card is LogSession, create the following URL:

`https://idp.amlab.net:8443/nidp/app/login?id=LogSession`

5 Send the URL of the LogSession card and the name of the ticket to the user.

Enabling a Logging Session

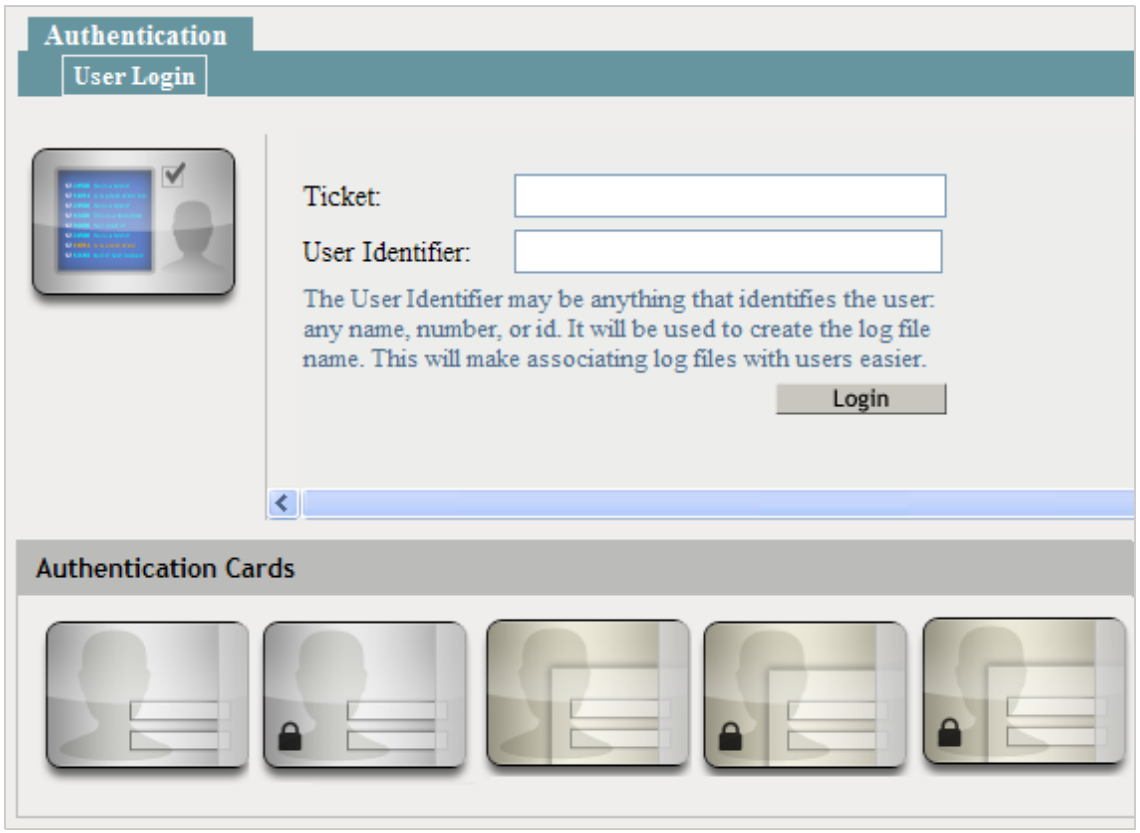
These steps are performed by the user. The URL needs to be sent to the user, with the ID and ticket values that were specified in "Creating a Logging Ticket" on page 857.

1 Open a browser and enter the log session URL sent by the help desk.

If the URL does not display a page that prompts for the ticket name, check the value of the id string. The id must be set to the ID of the authentication card of the Logging Session contract.

Instead of sending the user a URL, you can enable the **Show Card** option for the Logging Session card. When you do this, all users can see it. You need to decide if this is acceptable.

When the Show Card option is enabled, the login page looks similar to the following:



2 When prompted, enter the following:

Ticket: Specify the ticket name that the help desk sent.

User Identifier: Specify a value that the help desk associates with you as a user. The identifier must be less than 33 characters and contain only alphanumeric characters.

3 Click **Login**.

This login creates the logging session.

4 Enter your name and password, then click **Login**.

This login authenticates you to the Identity Server.

5 In the same browser window, enter the URL of the resource that is causing the problem.

6 Perform any other actions necessary to create the problem behavior.

7 Log out and send your user identifier to the help desk.

Viewing the Log File

These steps are performed by someone who has had Access Manager training and understands the significance of the messages in the log files. This can be an IDP Administrator or a specialist.

1 On the Identity Server, change to the Identity Server log directory.

`/var/opt/novell/nam/logs/idp/nidplogs`

2 Open the file that begins with the user identifier to which a session ID is appended.

If the user does not log out (just closes the browser window or the problem closes it), the session remains in the list of logged sessions. After 10 minutes of inactivity, the session is closed and the lock on the logging file is cleared. As long as the file is locked, no other application can read the file.

When a ticket expires, logging is stopped automatically. If you know that user is experiencing a problem that prevents the user from logging out, you might want to create a ticket with a short time limit.

3 (Conditional) If the user was experiencing a problem with an Embedded Service Provider, change to the Identity Server log directory on the device:

`/opt/novell/nam/webapps/nesp/WEB-INF/logs`

4 Open the file with the same user identifier and session ID.

5 After solving the problem, delete the file from each Identity Server in the cluster and each Access Gateway in the cluster.

Access Gateway Logging

- "Managing Access Gateway Logs" on page 860
- "Configuring Logging for a Proxy Service" on page 861

Managing Access Gateway Logs

In the Access Gateway, you can configure logging by using Advanced Options.

- "Configuring the Log Level" on page 860
- "Configuring the Log File" on page 861

Configuring the Log Level

1 In the Administration Console, select **Devices > Access Gateways > Edit > Advanced Options.**

2 Add the following line with appropriate log level:

```
LogLevel <loglevel>
```

Replace *loglevel* option with `emerg`, `alert`, `crit`, `error`, `warn`, `notice`, `info` or `debug`. The default log level is warn.

Option	Description
emerg	Sends only messages that render the system unusable, if they are not resolved.
alert	Sends only messages that require immediate action.
crit	Sends only messages about critical situations
error	Sends warning messages about recoverable errors.
warn	Sends warning messages.
Notice	Sends information about the status of a service to the service configuration logs.
Info	Sends informational messages such as requests sent to Web servers and the results of authentication requests.
Debug	Sends debug messages

3 Click **OK**.

4 Click Access Gateways, then click **Update** > **OK**.

The `error_log` file is available at `/var/opt/novell/nam/logs/mag/apache2/`.

Configuring the Log File

1 In the Admin console, click **Devices > Access Gateways > Edit > Advanced Options**

2 Add the following line:

`ErrorLog <path to the file where logs should be recorded>`

3 Click **OK**.

4 Click the Access Gateways link, then click **Update > OK**.

Configuring Logging for a Proxy Service

Logging HTTP transactions has associated costs. The Access Gateway is capable of handling thousands of transactions per second. If transaction volume is high and each log entry consumes a few hundred bytes, the Access Gateway can fill up the available disk space in a matter of minutes. HTTP logging also increases system overhead, which causes some degradation in performance. By default, the logging of HTTP transactions is turned off. Before enabling logging, you need to determine what needs to be logged and then plan a logging strategy. For more information about custom log formats, see Apache Log Configuration Module.

- “Determining Logging Requirements” on page 861
- “Calculating Rollover Requirements” on page 862
- “Enabling Logging” on page 864
- “Configuring Common Log Options” on page 864
- “Configuring Extended Log Options” on page 865
- “Configuring the Size of the Log Partition” on page 868

Determining Logging Requirements

Because logging requirements and transaction volume vary widely, NetIQ cannot make recommendations regarding a specific logging strategy. The following tasks guide you through the process of creating a strategy that fits your business needs.

1 Identify the reasons for tracking transactions such as customer billing, statistical analysis, or growth planning.

2 Determine which resources need logging.

You enable logging at the proxy service level. If you have a proxy service protecting resources whose transactions do not need to be logged, reconfigure your proxy services so that the proxy service you configure for logging contains only the resources for which you want to log transactions.

3 Determine what information you need in each log entry.

The common configuration for a log entry contains minimal information: the date, time, and client IP address for each entry. If you need more information, you can select the extended log configuration. Do not select all available fields, but carefully select what you really need. For example, you can include cookie information, but cookie information can consume a large amount of space and might not include any critical information you need.

You should log only the essential data because a few bytes can add up quickly when the Access Gateway is tracking thousands of hits every second. For information about what is available in an extended log profile, see “Configuring Extended Log Options” on page 865.

4 Design a rollover strategy.

A log must be closed before it can be downloaded to another server for analysis or deleted. You specify either by time or size when the Access Gateway closes a log file and creates a new one. For each proxy service that you enable for logging, you need to reserve enough space for at least two files: one for logging and one for rollover. To calculate the best procedure, see “Calculating Rollover Requirements” on page 862.

5 Design a log deletion strategy

The Access Gateway has a limited amount of disk space allocated for logging, and you need to decide how you are going to manage this space. You can limit the number of rollover files by number or age. To calculate the best procedure, see “Calculating Rollover Requirements” on page 862.

Calculating Rollover Requirements

You can have the Access Gateway roll over log files based on time or on size, but not both. If you already know which option you want to use, scan this section and then complete only the calculations pertinent to your choice. If you don’t know which option best matches your situation, completing the calculations in this section should help you decide.

The following variables are used in the formulas:

- **logpartition_size:** The total disk capacity reserved for log files on the Access Gateway.

  The Access Gateway reserves 4 GB to share between logging and system files. The system files do not grow significantly, so you can assume that you have about 2 GB for logging. To increase this size, see “Configuring the Size of the Log Partition” on page 868.

- **logentry_size:** The average log entry size.

  You can determine this by configuring a proxy service to track the required information, generating traffic to the proxy service, downloading the log files, determining how large each entry is, and calculating the average.

- **request_rate:** The peak rate of requests per second.

  You can estimate this rate or place your Access Gateway in service and get more accurate data by accessing generated statistics. See “Monitoring Access Gateway Statistics” on page 882.

- **num_services:** The number of proxy services for which you plan to enable logging.

- **logs_per_service:** The number of log files, both active and closed, that you want the Access Gateway to generate for each proxy service before the disk fills.

  You must plan to have at least two logs per proxy service, but you can have more.

The following formulas can help you estimate when the system would run out of resources:

- “Calculating diskfull_time” on page 862
- “Calculating max_roll_time” on page 863
- “Calculating max_log_roll_size” on page 863

Calculating diskfull_time

Use the following formula to calculate how long it takes the Access Gateway to fill your logging disk space:

```
diskfull_time in seconds = logpartition_size / (request_rate *
   logentry_size * num_services)
```

For example, assume the following:

logpartition_size = 1 GB (1,073,741,824 bytes)
request_rate = 1000 requests per second
logentry_size = 1 KB (1,024 bytes)
num_services = 1

```
diskfull_time = (1 GB) / (1000 * 1 KB * 1) = 1048 seconds (17.47
   minutes)
```

The logging disk space fills up every 17.47 minutes.

To calculate the diskfull_time for your Access Gateway:

1. Determine the values of the four variables listed above.
2. Use the diskfull_time formula to calculate how often you can expect your logging disk to fill, then use the result in Calculating max_roll_time.

If your diskfull_time interval is too short to be practical for your rollover schedule, the easiest option is to reduce the log entry size by configuring the proxy services to log less information per transaction.

Calculating max_roll_time

Use the following formula to calculate the maximum rollover time value you should specify in the **Roll over every** field

```
max_roll_time = diskfull_time / logs_per_service
```

For example, assume the following:

diskfull_time = 12 hours
logs_per_service = 2

```
max_roll_time = 12 / 2 = 6 hours
```

If you roll your logs over by time intervals, the maximum time should be less than six hours. Otherwise, scheduling the download and deletion of log files is much more complicated and the window in which this can be done is narrower.

To calculate the max_roll_time for your Access Gateway:

1. Determine how many log files you want the Access Gateway to generate per service before log space fills.

   The minimum number is two.
2. Use the max_roll_time formula and the diskfull_time value obtained in “Calculating diskfull_time” on page 862 to calculate how often you should have the cache device roll over the log files.
3. Record the max_roll_time result on your planning sheet.

Calculating max_log_roll_size

Use the following formula to calculate the maximum log file size you should specify in the **Maximum File Size** field:

```
max_log_roll_size = logpartition_size / (num_services *
   logs_per_service)
```

For example, assume the following:

logpartition_size = 600 MB
num_services = 2
logs_per_service = 3

```
max_log_roll_size = 600 MB / (2 * 3) = 100 MB
```

If you roll your logs over when they reach a specific size, the file size must be no more than 100 MB. Otherwise, the system runs out of disk space before you have three complete log files and scheduling the download and deletion of log files is much more complex.

To calculate the max_log_roll_size for your Access Gateway:

1. Determine the values of the three variables listed above.
2. Use the max_log_roll_size formula to calculate the maximum size a log file should reach before the cache device rolls it over.

Enabling Logging

Do not enable logging until you have designed a logging strategy. See “Determining Logging Requirements” on page 861.

1. In the Administration Console, click **Devices** > **Access Gateways** > **Edit** > **[Name of Reverse Proxy]** > **[Name of Proxy Service]** > **Logging.**
2. Fill in the following fields:

   Enable Logging: Select this field to enable logging.

   Log Directory: Default location for log files of proxy service is `/var/log/novell/reverse/<reverse_proxy_name>`.
3. In the **Logging Profile List**, click one of the following options:
   - **New:** Click this option to create a new logging profile. Then specify a name and select either **Common** or **Extended**.
   - **Default:** Click **Default** to modify or view the settings for the **Default** profile. The **Default** profile uses the common log options.

   A logging profile determines the type of information that is written to the log file; it also manages rollover and old file options.
4. Continue with one of the following:
   - “Configuring Common Log Options” on page 864
   - “Configuring Extended Log Options” on page 865

Configuring Common Log Options

Use the common log options page to control log rollover and old file options. The data included in a log entry is controlled by a default configuration that includes the following:

- Date and time of the request
- IP address of the client
- Remote host name
- The request line as it came from the client

- The HTTP status code returned to the client
- The number of bytes in the document transferred to the client

The Access Gateway does not allow active log files to be deleted. Only log files that have been closed can be deleted. The rollover options allow you to control when a file is rolled over and closed, and a new file is created. The old file options allow you to control when the rolled-over log files are deleted.

To configure a default log file for a selected proxy service:

1. Click **Devices > Access Gateways** > **Edit** > **[Name of Reverse Proxy]** > **[Name of Proxy Service]** > **Logging** > **[Name of Common Log Profile]**.
2. Select one of the following rollover options:

   Rollover When File Size Reaches: Rolls the file when it reaches the specified number of megabytes.

   Rollover every: Rolls the file at the specified interval. You can specify the interval in hours or days.

   - **beginning:** Specifies the day that the interval should begin. You can select a day of the week or the first of the month.
   - **at:** Select the hour of the day that the interval should begin and the time zone (either the local time zone or GMT).
3. Select one of the following old file options:

   Limit Number of Files to: Allows you to limit the number of old log files on the system to the number specified in this option. The oldest file is automatically deleted when this number is reached. All logging data in deleted files is lost.

   Delete Files Older Than: Allows you to configure the Access Gateway to delete files when they are older than the time you specify. All logging data in deleted files is lost.

   Do Not Delete: Prevents the system from automatically deleting the log files.
4. Click **OK**.
5. Click the **Access Gateways** link, then click **Update** > **OK**.

Configuring Extended Log Options

Use the extended log options page to control log entry content, log rollover, and old file options. A log entry always includes the date, time, and client IP address for each entry, but with the log data options, you can add other fields such as the IP address of the server and the username of the client.

The Access Gateway does not allow active log files to be deleted. Only log files that have been closed can be deleted. The rollover options allow you to control when a file is rolled over and closed, and a new file is created. The old file options allow you to control when the rolled-over log files are deleted.

To configure an extended log file for a selected proxy service:

1. Click **Devices** > **Access Gateways** > **Edit** > **[Name of Reverse Proxy]** > **[Name of Proxy Service]** > **Logging** > **[Name of Extended Log Profile]**.
2. Select one or more of the log data options:

Name	Description	Entry in Configuration File	Sample Entry in the Log file
User Name	The name of the user sending the request.	%u	"public”,”cn=admin,o=novell"
Server IP	The IP address of the Access Gateway.	%a	123.1.2.3
Site Name	The name of the reverse proxy.	%v	www.lagssl.com
Method	The HTTP method the browser sent to the Access Gateway.	%m	GET,POST
URI	The HTTP URL the browser sent to the Access Gateway.	%U	nam/acme_ss_js7.html
URI Stem	The stem portion of the HTTP URL the browser sent to the Access Gateway. The stem is everything in the URL up to the first question mark. If the URL has no question mark, the **URI Stem** field is the same as the **URI** field. **URI Stem** is redundant if **URI** is selected.		/path/to/resource
URI Query	The query portion of the HTTP URL the browser sent to the Access Gateway. The query is everything from the first question mark through the end of the URL. If the URL has no question mark, this field has no value. **URI Query** is redundant if URI is selected.	%q	?page=catalog&x=100&y=0
Version	The HTTP version specified in the URL the browser sent to the Access Gateway.		HTTP/1.1
Status	The HTTP status code the Access Gateway sent to the browser.	%s	200, 304, 404
Bytes Sent	The number of bytes of HTTP response data the Access Gateway sent to the browser.	%I	14378
Bytes Received	The number of bytes of HTTP request data the proxy service received from the browser.	%O	14378
Time Taken	The time it took the Access Gateway resources to deal with the request in microseconds.	%D	0.062, 0.392, 2, 802.1
User Agent	The User-Agent HTTP request header value the browser sent to the Access Gateway.	%{user-agent}i	Mozilla/5.0 (X11; Linux x86_64; rv:19.0) Gecko/20100101 Firefox/19.0

Name	Description	Entry in Configuration File	Sample Entry in the Log file
Cookie	The Cookie HTTP request header value the browser sent to the Access Gateway. The Access Gateway does not' cache cookie information. Cookies can consume a lot of space. If you select this option, make sure it contains the critical information that you need.	%{cookie}	IPCZQX0355730a2b=01001300a463874a93ef23e89e9acc94468beb4b; ZNPCQ003-37323400=c2e51552
Referer	The Referer HTTP request header value the browser sent to the Access Gateway.	%{Referer}	https://www.lagssl.com/netiq/nam/acme_ss_js7.html
Cached Status	The value indicates whether the request was filled from cache. 1 = filled from cache 0 = not filled from cache		0,1
Origin Server	The IP address of the Web server. This assumes the Access Gateway retrieved the requested information directly from the Web server.	%{BALANCER_WORKER_IP}e	125.1.2.5
X-Forward-For	The X-Forwarded-For HTTP request header value the browser sent to the Access Gateway. Do not confuse this with the X-Forwarded-For option, which causes the Access Gateway to generate or forward headers to upstream proxies or Web servers.	%{x-forward-for}i	10.0.0.1,10.0.02,10.0.03,10.0.0.4
Bytes Filled	The total bytes filled in response to the request.	%I	184
Content Range	The byte ranges sent from the Access Gateway to a requesting browser.	%{Content-Range}o	
E Tag	The tag sent from the Access Gateway to a requesting browser.	%{ETag}	604888-1077-466372c0
Completion Status	The completion status for the transaction, indicating that it completed successfully or that it failed. Possible values: success, timeout, reset (the client terminated the connection), administrative (the Access Gateway terminated the connection).	%X	success, timeout, reset
Reply Header Size	The size in bytes of the HTTP header associated with a response to a client.	%L	361
X Cache Info	Brief status statement for cached objects; brief reasons why an object was not cached.	%{Cache-Control}o	no-store

Name	Description	Entry in Configuration File	Sample Entry in the Log file
Range	The Range header value.	%{Range}o	
If Range	The If Range header value, which indicates whether the browser request was a conditional range request.	%{If-Range}	bytes 0-200/736
Content Length	The size in bytes of the entire object delivered to a requesting browser.	%O	741
Request Pragma	The pragma value associated with a browser request.	%{Pragma}o	No-cache , no-store
Reply Pragma	The pragma value associated with a server response to a requesting browser.	%{Pragma}i	no-cache

3 Select one of the following rollover options:

Rollover When File Size Reaches: Rolls the file when it reaches the specified number of megabytes.

Rollover every: Rolls the file at the specified interval. You can specify the interval in hours or days.

- **beginning:** Specifies the day that the interval should be begin. You can select a day of the week or the first of the month.
- **at:** Select the hour of the day that the interval should begin and the time zone (either the local time zone or GMT).

4 Select one of the following old file options:

Limit Number of Files to: Allows you to limit the number of old log files on the system to the number specified in this option. The oldest file is automatically deleted when this number is reached. All logging data in deleted files is lost.

Delete Files Older Than: Allows you to configure the Access Gateway to delete files when they are older than the time you specify. All logging data in deleted files is lost.

Do Not Delete: Prevents the system from automatically deleting the log files.

5 Click **OK**.

6 Click the **Access Gateways** link, then click **Update** > **OK**.

Configuring the Size of the Log Partition

The size of the log partition should be configured as part of the installation process. The Access Gateway logs are stored in the `/root` partition by default. You can create a `/var` partition to store the logs. The size of this partition depends on your requirements.

Downloading Log Files

The General Logging page displays the location of the files that Access Manager Appliance components use for logging system messages. There are some exceptions:

- **Default Auditing File:** If you have configured Novell Audit to send events to the default audit file (`/var/opt/novell/naudit/logs/auditlog`), this file does not appear in the list.

  If you want this file to appear in this list, you must make this file readable by the novlwww user. It is a breach of Novell Audit security for Access Manager code to change the permissions on this file. You must decide whether changing its permissions and displaying the file in this list compromises your security.

  To add this file in the list of files for the Administration Console, configure the following:

  - Use commands similar to the following to grant the `novlwww` user executable permissions to the naudit directories:

    ```
    chmod o+rx /var/opt/novell/naudit
    ```

    ```
    chmod o+rx /var/opt/novell/naudit/logs
    ```

  - Use a command similar to the following to grant the `novlwww` user read access to the `auditlog` file:

    ```
    chmod o+r /var/opt/novell/naudit/logs/auditlog
    ```

- **Proxy Service Log Files:** If you enable proxy service logging, these files are not available for downloading from this page because there could be potentially hundreds of these files. If this type of logging has been enabled, the directory where they are located is displayed. For more information about this type of logging, see “Configuring Logging for a Proxy Service” on page 861.

To view or download a log file:

1. Click **Auditing** > **General Logging**.
2. Select one or more log files, click **Download**, then open it or save it to disk.

   You can use any text editor to view the file.

NOTE: The central location of all log files is `/var/opt/novell/nam`.

Each Access Manager Appliance component generates multiple log files. The following tables lists these files and the types of messages they contain:

- “Administration Console Logs” on page 870
- “Identity Server Logs” on page 870
- “Access Gateway Appliance and Access Gateway Service Logs” on page 870

Administration Console Logs

Filename	Description
/var/opt/novell/nam/logs/adminconsole/tomcat/catalina.out	Contains Tomcat errors.
/var/opt/novell/nam/logs/adminconsole/volera/app_sc.0.log	Contains events related to importing devices, device configuration changes, health status changes, statistics reporting, and communication problems.
/var/opt/novell/nam/logs/adminconsole/volera/app_cc.0.log	Contains events related to policy configuration.
/var/opt/novell/nam/logs/adminconsole/volera/platform.0.log	Contains XML events for configuration changes. This log file contains very little useful information for system administrators.

Identity Server Logs

Filename	Description
/var/opt/novell/nam/logs/idp/tomcat/catalina.out	Logging to this file occurs only if you have selected the **Echo to Console** option from the **Identity Servers** > **Servers** > **Edit** > **Logging** page. When component logging has been set to info for Applications, it contains entries tracing user authentication and role assignments.
/var/opt/novell/nam/logs/jcc/jcc-0.log.0	Contains the log entries for the server communications module related to interaction of the Identity Server with the Administration Console, such as imports, certificates, health checks, and configuration.

Access Gateway Appliance and Access Gateway Service Logs

Filename	Description
/var/opt/novell/nam/logs/mag/tomcat/catalina.out	Logging to this file only occurs if you have selected the **Echo to Console** option from the **Identity Servers** > **Servers** > **Edit** > **Logging** page. Check this file for entries tracing the evaluation of authorization, identity injection, and form fill policies.
/var/log/novell/reverse/<proxy_service-name>	If logging is enabled on one or more reverse proxies, this directory contains the log files. A directory is listed for each reverse proxy on which you have enabled logging.

Filename	Description
/var/opt/novell/nam/logs/jcc/jcc-0.log.0	Contains the log entries for the server communications module related to interaction of the Access Gateway with the Administration Console, such as imports, certificates, health checks, and configuration.
/var/opt/novell/nam/logs/mag/apache2/error_log This directory also contains the Apache generated log files such as the `error_log` file.	
/var/opt/novell/nam/logs/mag/amlogging/ags_error.log	Contains the messages generated for configuration, device imports, health, and statistics. It also contains entries for the policy evaluation processes done by the Gateway Service Manager module.
/var/opt/novell/nam/logs/mag/amlogging/verbose_log	Contains the verbose log entries.

Turning on Logging for Policy Evaluation

Policy evaluation for roles occurs at the Identity Server. For Authorization and Identity Injection policies, policy evaluation occurs on the Embedded Service Provider (ESP) where the policy is enabled.

For the Form Fill policies, the evaluation and logging is done by ESP and the proxy service. To set the logging level on the Access Gateway for the proxy service, see “Enabling Form Fill Logging” on page 1041.

Logging for the policy evaluation done by ESP is controlled by the log settings of the Identity Server configuration. To enable this type of logging:

1. Click **Devices > Identity Servers** > ***Edit*** > **Logging**.

   If you have set up more than one Identity Server configuration, make sure you select the configuration to which the other Access Manager Appliance components have been assigned.
2. Select **Enabled** for **File Logging**.
3. Select to echo the trace messages to the console: For the Access Gateway Appliance, Access Gateway Service, or Identity Server, this sends the messages to the `catalina.out` file.
4. (Optional) Specify a path for the Identity Server log files.
5. For policy evaluation tracing, set the **Application** level to **info** in the **Component File Logger Levels** section.

   If you are only troubleshooting policies at this time, do not select any other options. This reduces the amount of information recorded in the log files.

   To see the policy SOAP messages, you need to set the **Application** level to **config.**
6. Update the Identity Server.
7. Click **Auditing** > **General Logging**.
   - For role evaluation traces, view the Identity Server `catalina.out` file.

     If your Identity Servers are clustered, you need to look at the file from each Identity Server.
   - For Authorization, Form Fill, and Identity Injection evaluation traces, view the log file of ESP of the device that is protecting the resource.

Access Gateway Appliance or Service: This is the `catalina.out` file of the Access Gateway where the protected resource is defined. If the Access Gateway is part of a cluster, you need to look at this file from each Access Gateway in the group.

To view the actual ESP log file that contains only ESP log messages, see the `nidp.*.xml` files in the `/var/opt/novell/tomcat7/webapps/nesp/WEB-INF/logs` directory (or the directory you specified in Step 4). Depending upon how you have configured **File Wrap**, the * portion of the filename contains the month, the week, the day, and the hour.

8 To understand what you are looking for in the log file, continue with one of the following:

- "Understanding Policy Evaluation Traces" on page 1028 if you set **Application** level to **info.**
- "Policy Evaluation: Access Gateway Devices" on page 1011 if you set **Application** level to **config**.

19 Monitoring Component Statistics

The Statistics page allows you to monitor the amount of data and the type of data that Identity Server and Access Gateway processes. You can specify the intervals for the refresh rate and, where allowed, view graphic representations of the activity.

- “Identity Server Statistics” on page 873
- “Access Gateway Statistics” on page 882
- “Component Statistics Through REST APIs” on page 893

Identity Server Statistics

- “Monitoring Identity Server Statistics” on page 873
- “Monitoring Identity Server Cluster Statistics” on page 881

Monitoring Identity Server Statistics

1. In the Administration Console, choose **Devices > Identity Servers**.
2. In the **Statistics** column, click **View**.
3. Click either of the following options:

   Statistics: Select this option to view the statistics as currently gathered. The page is static and the statistics are not updated until you click **Live Statistics Monitoring**.

   Live Statistics Monitoring: Select this option to view the statistics as currently gathered and to have them refreshed at the rate specified in the **Refresh Rate** field.
4. Review the following statistics:

   - “Application” on page 874
   - “Authentications” on page 874
   - “Incoming HTTP Requests” on page 875
   - “Outgoing HTTP Requests” on page 875
   - “Liberty” on page 876
   - “SAML 1.1” on page 876
   - “SAML 2” on page 877
   - “WSF (Web Services Framework)” on page 877
   - “Clustering” on page 879
   - “LDAP” on page 880
   - “SP Brokering” on page 880
   - “Risk-Based Authentication” on page 881

5. Click **Close** to return to the Servers page.

NOTE: The statistics graphs of the Identity Server and Access Gateway are available only in the primary Administration Console. The periodic stats are sent to the secondary Administration Console only when the primary console is not available. Hence, the statistics graphs of the Identity Server and Access Gateway do not display any statistics values in the secondary Administration Console.

Application

Statistic	Description
Free Memory	The percentage of free memory available to the JVM (Java Virtual Machine). Click **Graphs** to view memory usage for a specific unit of time (1 hour, 1 day, 1 week, 1 month, 6 months, or 12 months). The Value axis displays the percentage of memory that is free for the selected time period.

Authentications

Statistic	Description
Provided Authentications	The number of successful provided authentications given out to external entities after the Identity Server was started.
Consumed Authentications	The number of successful consumed authentications after the Identity Server was started.
Provided Authentication Failures	The number of failed provided authentications given out to external entities after the Identity Server was started.
Consumed Authentication Failures	The number of failed consumed authentications after the Identity Server was started. The Consumed Authentication Failures statistics counter increases by one whenever a method execution fails for the user.
Historical Maximum Logins Served	The maximum number of logins served during an interval and displayed after completion of the interval.
Logins In Last Interval	The number of active user sessions during the last interval.
Logouts	The number of explicit logouts performed by users. This does not include logouts where an inactive session was destroyed.
Cached Sessions	The number of currently active cached user sessions. This represents the number of users currently logged into the system; however, if a single person has two browser windows open on the same client and if that person performed two distinct authentications, then that person has two user sessions. Click **Graphs** to view the number of cached sessions for a specific unit of time (1 hour, 1 day, 1 week, 1 month, 6 months, or 12 months). The Value axis displays the number of cached sessions. If no sessions have been cached, the value axis is not meaningful.
Cached Ancestral Sessions	The number of cached ancestral session IDs. An ancestral session ID is created during the failover process. When failover occurs, a new session is created to represent the previous session. The ID of the previous session is called an “ancestral session ID,” and it is retained for subsequent failover operations.
Cached Subjects	The number of current cached subject objects. Conceptually, the cached subjects are identical to the cached principals.

Statistic	Description
Cached Principals	The number of current cached principal objects. A principal can be thought of as a single directory user object. Multiple users can log in using a single directory user object, in which case multiple cached sessions would exist sharing a single cached principal.
Cached Artifacts	The number of current cached artifact objects. During authentication, an artifact is generated that maps to an assertion. This cache holds the artifact to assertion mapping until the artifact resolution request is received. Under normal operations, artifacts are resolved within milliseconds of being placed in this cache.

Incoming HTTP Requests

Incoming HTTP requests are divided into three categories: active, interval, and historical. As soon as a request is complete, it is placed into the interval category. The interval represents the last 60 seconds of processed requests. At the completion of the 60-second interval, all requests in the interval category are merged into the historical category.

Statistic	Description
Total Requests	The total number of incoming HTTP requests that have been processed after the Identity Server was started. Click **Graphs** to view the number of requests for a specific unit of time (1 hour, 1 day, 1 week, 1 month, 6 months, or 12 months). The Value axis displays the number of requests for the selected time period.
Currently Active Requests	The number of currently active incoming HTTP requests.
Oldest Active Request (Milliseconds)	The age of the oldest currently active incoming HTTP request.
Last Interval Maximum Request Duration (Milliseconds)	The age of the longest incoming HTTP requests that was processed during the last 60-second interval.
Last Interval Mean Request Duration (Milliseconds)	The mean age of all incoming HTTP request that were processed during the last 60-second interval.
Historical Maximum Request Duration (Milliseconds)	The age of the longest incoming HTTP request that was processed after the Identity Server was started.
Historical Mean Request Duration (Milliseconds)	The mean age of all incoming HTTP requests that were processed after the Identity Server was started.

Outgoing HTTP Requests

Outgoing HTTP requests are divided into three categories: active, interval, and historical. As soon as a request is complete, it is placed into the interval category. The interval represents the last 60 seconds of processed requests. At the completion of the 60-second interval, all requests in the interval category are merged into the historical category.

Statistic	Description
Total Requests	The total number of outgoing HTTP requests that have been processed after the Identity Server was started. Click **Graphs** to view the number of requests for a specific unit of time (1 hour, 1 day, 1 week, 1 month, 6 months, or 12 months). The Value axis displays the number of requests for the selected time period.
Currently Active Requests	The number of currently active outgoing HTTP requests.
Oldest Active Request (Milliseconds)	The age of the oldest currently active outgoing HTTP request.
Last Interval Maximum Request Duration (Milliseconds)	The age of the longest outgoing HTTP request that was processed during the last 60-second interval.
Last Interval Mean Request Duration (Milliseconds)	The mean age of all outgoing HTTP requests that were processed during the last 60-second interval.
Historical Maximum Request Duration (Milliseconds)	The age of the longest outgoing HTTP request that was processed after the Identity Server was started.
Historical Mean Request Duration (Milliseconds)	The mean age of all outgoing HTTP requests that were processed after the Identity Server was started.

Liberty

Statistic	Description
Liberty Federation	The number of Liberty protocol federations performed after the Identity Server was started.
Liberty De-Federations	The number of Liberty protocol defederations performed after the Identity Server was started.
Liberty Register-Names	The number of Liberty protocol register names performed after the Identity Server was started.

SAML 1.1

Statistic	Description
SAML1.1 Attribute Queries	The number of SAML 1.1 protocol attribute queries performed after the Identity Server was started.

SAML 2

Statistic	Description
SAML2 Attribute Queries	The number of SAML 2 protocol attribute queries performed after the Identity Server was started.
SAML2 Federations	The number of SAML 2 protocol federations performed after the Identity Server was started.
SAML2 Defederations	The number of SAML 2 protocol defederations performed after the Identity Server was started.
SAML2 Register-Names	The number of SAML 2 protocol register names performed after the Identity Server was started.

WSF (Web Services Framework)

Statistic	Description
Personal Profile Service Queries	The number of Liberty IDSIS Personal Profile Web Service queries performed after the Identity Server was started.
Personal Profile Service Modifies	The number of Liberty IDSIS Personal Profile Web Service changes performed after the Identity Server was started.
Employee Profile Service Queries	The number of Liberty IDSIS Employee Profile Web Service queries performed after the Identity Server was started.
Employee Profile Service Modifies	The number of Liberty IDSIS Employee Profile Web Service changes performed after the Identity Server was started.
Custom Profile Service Queries	The number of Novell Custom Profile Web Service queries performed after the Identity Server was started.
Custom Profile Service Modifies	The number of Novell Custom Profile Web Service changes performed after the Identity Server was started.
Credential Profile Service Queries	The number of Novell Credential Profile Web Service queries performed after the Identity Server was started.
Credential Profile Service Modifies	The number of Novell Credential Profile Web Service changes performed after the Identity Server was started.
Authentication Profile Service Queries	The number of Novell Authentication Profile Web Service queries performed after the Identity Server was started.
Authentication Profile Service Modifies	The number of Novell Authentication Profile Web Service changes performed after the Identity Server was started.
LDAP Profile Service Queries	The number of Novell LDAP Profile Web Service queries performed after the Identity Server was started.
LDAP Profile Service Modifies	The number of Novell LDAP Profile Web Service changes performed after the Identity Server was started.
Constant Profile Service Queries	The number of Novell Constant Profile Web Service queries performed after the Identity Server was started.

Statistic	Description
Discovery Service Queries	The number of Liberty Discovery Web Service queries performed after the Identity Server was started.
Discovery Service Modifies	The number of Liberty Discovery Web Service changes performed after the Identity Server was started.
Redirected Interaction Service Requests	The number of Liberty User Interaction Redirection Profile requests performed after the Identity Server was started.
Trusted Interaction Service Requests	The number of Liberty User Interaction Trusted Service Profile requests performed after the Identity Server was started.
Client of Redirected Interaction Service Requests	The number of Liberty User Interaction Redirection Profile requests initiated as a client after the Identity Server was started.
Client of Trusted Interaction Service Requests	The number of Liberty User Interaction Trusted Service Profile requests initiated as a client after the Identity Server was started.
Data Location LDAP	The number of attempts to use LDAP as a data location for a query or a modify of any Web Service after the Identity Server was started.
Data Location LDAP Aggregation	The number of attempts to use LDAP as a data location for aggregation of a query or a modify of any Web Service after the Identity Server was started.
Data Location User Profile	The number of attempts to use the User Profile object as a data location for a query or a modify of any Web Service after the Identity Server was started. A User Profile object is a directory object stored in the Identity Server's configuration datastore.
Data Location User Profile Aggregation	The number of attempts to use the User Profile object as a data location for aggregation of a query or a modify of any Web Service after the Identity Server was started. A User Profile object is a directory object stored on the Identity Server's configuration datastore.
Data Location Remote	The number of attempts to use the Remote location as a data location for a query or a modify of any Web Service after the Identity Server was started. A Remote location includes Pushed Attributes and External Services.
Data Location Pushed Attributes	The number of attempts to use the Pushed Attributes as a remote data location for a query or a modify of any Web Service after the Identity Server was started.
Data Location Pushed Attributes Aggregation	The number of attempts to use the Pushed Attributes as an remote data location for aggregation of a query or a modify of any Web Service after the Identity Server was started.
Data Location External Service	The number of attempts to use an External Service as a remote data location for a query or a modify of any Web Service after the Identity Server was started. An External Service is where the same Web Service exists on an external Service Provider and a call can be made to request data from the service.

Clustering

An authoritative server is the cluster member that holds the authentication information for a given user session. For a request associated with a given session to be processed, it must be routed ("proxied") to the authoritative cluster member. If an L4 switch causes a request to go to a non-authoritative cluster member, that cluster member proxies the request to the authoritative cluster member.

When a request is received, a cluster member uses multiple means to determine which cluster member is the authoritative server for the request. It looks for a parameter on the query string of the URL indicating the authoritative server. It looks for an HTTP cookie, indicating the authoritative server. If these do not exist, the cluster member examines the payload of the HTTP request to determine the authoritative server. Payload examinations result in immediate identification of the authoritative server or a user session ID or user identity ID that can be used to locate the authoritative server.

If a user session ID or user identity ID is found, the ID is broadcast to all cluster members asking which member is the authoritative server for the given ID. The authoritative server receives the broadcast message, determines that it indeed holds the given session or user, and responds accordingly.

The higher the number of proxied requests, the lower the performance of the entire system. Furthermore, the higher the number of payload examinations and ID broadcasts, the lower the performance of the entire system. If these numbers are high, verify the configuration of the L4 switch. Ensure that the session persistence option is enabled, which allows clients to be directed to the same Identity Server after they have established a session.

Statistic	Description
Currently Active Proxied Requests	The number of currently active proxied HTTP requests.
Total Proxied Requests	The total number of proxied requests that have been processed after the Identity Server was started. A request becomes a proxied request when the request is sent first to a non-authoritative machine.
Total Non-Proxied Requests	The total number of non-proxied requests that have been processed after the Identity Server was started. A request becomes a non-proxied request when the request is sent first to the authoritative machine.
Authoritative Server Obtained from URL Parameter	The total number of authoritative servers identified by using the parameter from the URL query string after the Identity Server was started.
Authoritative Server Obtained from Cookie	The total number of authoritative servers identified by using the HTTP cookie after the Identity Server was started.
Payload Examinations	The total number of attempted payload examinations to identify the authoritative server after the Identity Server was started.
Successful Payload Examinations	The total number of successful payload examinations to identify the authoritative server after the Identity Server was started.
Identity ID Broadcasts	The total number of attempted Identity ID Broadcasts to identify the authoritative server after the Identity Server was started.
Successful Identity ID Broadcasts	The total number of successful Identity ID Broadcasts to identify the authoritative server after the Identity Server was started.
Session ID Broadcasts	The total number of attempted Session ID Broadcasts to identify the authoritative server.

Statistic	Description
Successful Session ID Broadcasts	The total number of successful Session ID Broadcasts to identify the authoritative server after the Identity Server was started.

LDAP

Statistic	Description
User Store Replica Restarts	The number of times that a user store replica became unavailable so that a restart was necessary after the Identity Server was started. A user store restart is attempted once every minute.
Successful User Store Replica Restarts	The number of times that a user store replica restart was successfully completed after the Identity Server was started.
User Store Replica Restart Retries	The number of times that a user store replica restart failed and was put back into “wait mode” to try again in one minute after the Identity Server was started.
Currently Active Connection Waits	The current number of user threads waiting for an LDAP connection to become available.
Connection Waits	The number of times that a user thread was required to wait for an LDAP connection to become available after the Identity Server was started. A wait would be required if the maximum number of connections allocated to the associated connection pool were all currently in use by other threads.
Connection Waits Aborted Due To Timeout	The number of times that an LDAP connection wait terminated because of the Identity Server timing out after the Identity Server was started. This would result in an LDAP Service Not Available error.
Connection Waits Aborted Due To Closed Pool	The number of times that an LDAP connection wait terminated because of a closed connection pool after the Identity Server was started. This would normally be caused by an LDAP replica failing while the user thread is waiting for the connection. This would result in an LDAP Service Not Available error.

SP Brokering

Statistic	Description
Total Brokering Requests	The total number of brokering requests created after the Identity Server was started. This count is a sum of all connections created to all replicas of the configuration datastore and all user stores.
Total Brokering Requests Denied Due to Group Check	The total number of brokering authentication requests denied in a target service provider. The brokering group can either be the identity provider or target service provider but both does not belong to the same group.
Total Brokering Requests Denied Due to Role Deny	The total number of brokering authentication requests to a target service provider denied due to broker policy evaluation denying the role.
Total Brokering Requests Passed	The total number of brokering requests passed after the Identity Server was started.

Risk-Based Authentication

Statistic	Description
Requests Allowed After Authentication	The total number of low risk requests allowed after authentication.
Requests Denied After Authentication	The total number of high risk requests denied after authentication.
Requests Allowed Pre-Authentication	The total number of low risk requests allowed during pre-authentication.
Requests Denied Pre-Authentication	The total number of high risk requests denied during pre-authentication.
Requests for Additional Authentication in Pre-Authentication	The total number of additional authentication requests during pre-authentication.
Requests for Additional Authentication in Post-Authentication	The total number of additional authentication requests during post-authentication.

Monitoring Identity Server Cluster Statistics

On the Cluster Statistics page, you can configure the list of statistics to show the desired statistics for an Identity Server cluster. See “Monitoring Identity Server Statistics” on page 873 for the complete list of statistics for each server in an Identity Server cluster.

To configure the statistics, perform the following steps:

1. In the Administration Console, click **Devices** > **Identity Servers** > **[Name of Cluster]** > **Statistics** > **Configure**.
2. Select **Set default statistics under "Selected Statistics"** to replace the selected statistics with the default statistics.

   The default statistics include Free Memory, Provided Authentications, Cached Sessions, Logins In Last Interval, Currently Active Requests - Incoming HTTP Requests, Currently Active Requests - Outgoing HTTP Requests, and Currently Active Proxied Requests.
3. Select statistics from **Available Statistics** and move to **Selected Statistics**.
4. Click **OK**.

   The Cluster Statistics page displays the summary of configured statistics for each individual member of the cluster.

   To view additional information about a specific Identity Server, click the name of the Identity Server in the **Server Name** column of the summary.

   You can also view all the statistics for an individual server of the cluster. Click **View** in the **Statistics** column of the summary to see these additional statistics. For more information, see “Monitoring Identity Server Statistics” on page 873.
5. Click **Close**.

Access Gateway Statistics

- “Monitoring Access Gateway Statistics” on page 882
- “Monitoring Access Gateway Cluster Statistics” on page 891

Monitoring Access Gateway Statistics

1 In the Administration Console, click **Devices** > **Access Gateways** > **[Name of Server]** > **Statistics**.

2 Select from the following types:

- “Server Activity Statistics” on page 882
- “Server Benefits Statistics” on page 886
- “Service Provider Activity Statistics” on page 887

3 Click **Close**.

NOTE: The statistics graphs of the Identity Server and Access Gateway are available in only the primary Administration Console. The periodic stats are sent to the secondary Administration Console only when the primary console is not available. Hence, the statistics graphs of the Identity Server and Access Gateway do not display any statistics values in the secondary Administration Console.

Server Activity Statistics

Select whether to monitor live or static statistics:

Statistics: Select this option to view the statistics as currently gathered. The page is static and the statistics are not updated until you click **Live Statistics Monitoring**.

Live Statistics Monitoring: Select this option to view the statistics as currently gathered and to have them refreshed at the rate specified in the **Refresh Rate** field.

These general statistics are grouped into the following categories:

- “Server Activity” on page 883
- “Connections” on page 884
- “Bytes” on page 884
- “Requests” on page 885
- “Cache Freshness” on page 886

Server Activity

The Server Activity section displays general server utilization statistics.

Statistic	Description
CPU Utilization	Displays the current CPU utilization rate. Use the available graph for capacity planning. Click **Graphs** to view the CPU usage for a specific unit of time (1 hour, 1 day, 1 week, 1 month, 6 months, or 12 months). The Value axis displays the percentage of use.
Cache Hit	Displays the current cache hit rate. A high cache hit rate indicates that the caching system is off-loading significant request processing from the Web servers whose objects have been cached. Click **Graphs** to view the number of cache hits for a specific unit of time (1 hour, 1 day, 1 week, 1 month, 6 months, or 12 months). The Value axis displays the number of hits.
Mounted Partitions Disk Space	Displays the total disk space configured on mounted partitions.
Mounted Partitions Disk Space Used	Displays the disk space in use on mounted partitions.
Mounted Partitions Disk Space Free	Displays the disk space available on mounted partitions.
Swap Partition Disk Space	Displays the total disk space configured for the swap partition. The Linux Gateway Service displays the available swap space reported by the Linux kernel (see sysinfo for details).
Swap Partition Disk Space Used	Displays the disk space in use on the swap partition.
Swap Partition Disk Space Free	Displays the disk space available on the swap partition.
Cache Disk Space	Displays the total disk space available for caching.
Cache Disk Space Utilization	Reserved. Not currently used.
Total Installed Memory	Displays the amount of memory that is installed on the Access Gateway.
Start Up Time	Displays the last time the Access Gateway was started.
Up Time	Displays the total time the Access Gateway has been running since it was last started.
Number of Objects Cached	Displays the total number of objects that have been cached since the Access Gateway was last started.

Connections

The connection statistics show the current and peak levels of usage in terms of TCP connections.

Statistic	Description
Current Connections to Origin Server	Displays the current number of connections that the Access Gateway has established with Web servers.
Current Connections to Browsers	Displays the current number of connections that the Access Gateway has established with browsers.
Current Total Connections	Displays the current total of all connections that the Access Gateway has established.
Connections to Origin Server	Displays the total number of connections that the Access Gateway has established with Web servers since it was last started. Click **Graphs** to view the number of connections for a specific unit of time (1 hour, 1 day, 1 week, 1 month, 6 months, or 12 months). The Value axis displays the number of connections.
Peak Connections from Origin Server	Displays the peak number of connections that the Access Gateway has established with Web servers.
Connections to Browsers	Displays the total number of connections that the Access Gateway has established with browsers since it was last started. Click **Graphs** to view the number of connections for a specific unit of time (1 hour, 1 day, 1 week, 1 month, 6 months, or 12 months). The Value axis displays the number of connections.
Peak Connections to Browsers	Displays the peak number of connections that the Access Gateway has established with browsers.
Total Connections through SOCKS	Displays the total number of connections the Access Gateway has established through a firewall.
Failed Connection Attempts	Displays the total number of failed connection attempts the Access Gateway has made while attempting to fill its Web object cache.

Bytes

The bytes statistics show how fast information is being sent in response to the following types of requests:

- Browser requests to the Access Gateway
- Access Gateway requests to the Web servers

Statistic	Description
Throughput of the Origin Server	Displays the average number of bytes of data being sent each second from the Web servers to the Access Gateway. Average number of bytes = total number of bytes sent from origin server to the Access Gateway per system uptime in seconds. Click **Graphs** to view the number of bytes for a specific unit of time (1 hour, 1 day, 1 week, 1 month, 6 months, or 12 months). The Value axis displays the number of bytes.

Statistic	Description
Throughput of the Browser	Displays the average number of bytes of data being sent each second from the Access Gateway to the browsers. Average number of bytes = total number of bytes sent from the Access Gateway to browsers per system uptime in seconds. Click **Graphs** to view the number of bytes for a specific unit of time (1 hour, 1 day, 1 week, 1 month, 6 months, or 12 months). The Value axis displays the number of bytes.
Total Bytes per Second	Displays the total number of bytes of data being sent each second from the Access Gateway and from the Web servers. Click **Graphs** to view the number of bytes for a specific unit of time (1 hour, 1 day, 1 week, 1 month, 6 months, or 12 months). The Value axis displays the number of bytes.
Bytes Sent to Origin Server	Displays the total number of bytes sent to the origin server after the server is started.
Bytes Received from Origin Server	Displays the total number of bytes of data sent to the Access Gateway from the Web servers since the Access Gateway last started.
Bytes Sent to Browser	Displays the total number of bytes of data sent to the browsers from the Access Gateway since the Access Gateway last started.
Bytes Received from Browser	The total number of bytes received from the browser after the server is started.
Total Bytes	Displays the total number of bytes of data sent from the Access Gateway and from the Web servers since the Access Gateway was last started.

Requests

The request statistics show the number of requests that are being sent from the browsers to the Access Gateway and from the Access Gateway to the Web servers.

Statistic	Description
Current Requests to Origin Server	Displays the current number of requests that the Access Gateway has made to the Web servers. Click **Graphs** to view the number of requests for a specific unit of time (1 hour, 1 day, 1 week, 1 month, 6 months, or 12 months). The Value axis displays the number of requests.
Current Requests from Browsers	Displays the current number of requests that the browsers have made to the Access Gateway. Click **Graphs** to view the number of requests for a specific unit of time (1 hour, 1 day, 1 week, 1 month, 6 months, or 12 months). The Value axis displays the number of requests.
Total Current Requests	Displays the total number of current requests that the Access Gateway has received from the browsers and that the Access Gateway has sent to the Web servers.
Successful Requests to Origin Server	Displays the total number of successful requests that the Access Gateway has sent to the Web servers since the Access Gateway last started.

Statistic	Description
Failed Requests to Origin Server	Displays the total number of failed requests that the Access Gateway has sent to the Web servers since the Access Gateway last started.
Cumulative Requests to Browsers	Displays the total number of requests that the browsers have sent to the Access Gateway since the Access Gateway last started.
Total Cumulative Requests	Displays the total number of cumulative requests that the Access Gateway has processed since the Access Gateway last started.
Requests per Second to Origin Server	Displays the number of requests that are being sent each second from the Access Gateway to the Web servers. Click **Graphs** to view the number of requests for a specific unit of time (1 hour, 1 day, 1 week, 1 month, 6 months, or 12 months). The Value axis displays the number of requests.
Requests per Second from Browsers	Displays the number of requests that are being sent each second from the browsers to the Access Gateway. Click **Graphs** to view the number of requests for a specific unit of time (1 hour, 1 day, 1 week, 1 month, 6 months, or 12 months). The Value axis displays the number of requests.
Total Requests per Second	Displays the total number of requests that are being sent each second from the Access Gateway and from the browsers.
Peak Requests per Second to Origin Server	Displays the peak number of requests that have been sent in one second from the Access Gateway to the Web servers.
Peak Requests per Second from Browsers	Displays the peak number of requests that have been sent in one second from the browsers to the Access Gateway.

Cache Freshness

The cache freshness statistics display information about the cache refresh process.

Statistic	Description
Total “Get If Modified Since” Request	Displays the total number of Get If Modified Since requests that the Access Gateway has received from browsers.
Total Not Modified Replies	Displays the total number of 304 Not Modified replies that the Access Gateway has received from the Web servers for updated content.
Cache Freshness	Displays the percentage of objects in cache that are considered fresh. Click **Graphs** to view the percentage of fresh objects for a specific unit of time (1 hour, 1 day, 1 week, 1 month, 6 months, or 12 months). The Value axis displays the percentage of fresh objects.
Oldest Object in Memory	Displays how long the oldest cache object has been cached.

Server Benefits Statistics

Select whether to monitor live or static statistics:

Statistics: Select this option to view the statistics as currently gathered. The page is static and the statistics are not updated until you click **Live Statistics Monitoring**.

Live Statistics Monitoring: Select this option to view the statistics as currently gathered and to have them refreshed at the rate specified in the **Refresh Rate** field.

The Server Benefits page displays information about bandwidth and DNS caching:

Statistic	Description
Total Bandwidth Saved	Displays the amount of bandwidth saved by using data cached by the Access Gateway rather than requesting the data from the Web servers.
Bytes Saved per Second	Displays how many bytes of data the Access Gateway was able to send from cache rather than requesting it from the Web servers.

Service Provider Activity Statistics

Select whether to monitor live or static statistics:

Statistics: Select this option to view the statistics as currently gathered. The page is static and the statistics are not updated until you click **Live Statistics Monitoring**.

Live Statistics Monitoring: Select this option to view the statistics as currently gathered and to have them refreshed at the rate specified in the **Refresh Rate** field.

The ESP Activity page displays information about the communication process between the Access Gateway (ESP) and the Identity Server. These statistics are grouped into the following categories:

- Application
- Authentications
- Incoming HTTP Requests
- Outgoing HTTP Requests
- Liberty
- Clustering
- SP Brokering

Click **Graphs** to review historical statistics.

Application

Statistic	Description
Free Memory	The percentage of free memory available to the JVM (Java Virtual Machine). Click **Graphs** to view the free memory for a specific unit of time (1 hour, 1 day, 1 week, 1 month, 6 months, or 12 months). The Value axis displays the percentage of free memory.

Authentications

Statistic	Description
Provided Authentications	The number, since the Identity Server was started, of successful provided authentications given out to external entities.

Statistic	Description
Consumed Authentications	The number, since the Identity Server was started, of successful consumed authentications.
Provided Authentication Failures	The number, since the Identity Server was started, of failed provided authentications given out to external entities.
Consumed Authentication Failures	The number, since the Identity Server was started, of failed consumed authentications. **NOTE:** The consumed authentication failures does not show the number of invalid password attempt failures of the Identity Provider in the statistics page.
Historical Maximum Logins Served	The maximum number of logins served during an interval and displayed after completion of the interval.
Logins in Last Interval	The number of active user sessions during the last interval.
Logouts	The number of explicit logouts performed by users. This does not include logouts where an inactive session was destroyed.
Cached Sessions	The number of currently active cached user sessions. This represents the number of users currently logged into the system with the following caveat: If a single person has two browser windows open on the same client and if that person performed two distinct authentications, then that person has two user sessions. Click **Graphs** to view the number of cached sessions for a specific unit of time (1 hour, 1 day, 1 week, 1 month, 6 months, or 12 months). The Value axis displays the number of cached sessions. If no sessions have been cached, the value axis is not meaningful.
Cached Ancestral Sessions	The number of cached ancestral session IDs. An ancestral session ID is created during the failover process. When failover occurs, a new session is created to represent the previous session. The ID of the previous session is termed an “ancestral session ID,” and it is persisted for subsequent failover operations.
Cached Subjects	The number of current cached subject objects. Conceptually, the cached subjects are identical to the cached principals.
Cached Principals	The number of current cached principal objects. A principal can be thought of as a single directory user object. Multiple users can log in using a single directory user object, in which case multiple cached sessions would exist sharing a single cached principal.
Cached Artifacts	The number of current cached artifact objects. During authentication, an artifact is generated that maps to an assertion. This cache holds the artifact to assertion mapping until the artifact resolution request is received. Under normal operations, artifacts are resolved within milliseconds of being placed in this cache.

Incoming HTTP Requests

Incoming HTTP requests are divided into three categories: active, interval, and historical. As soon as a request is complete, it is placed into the interval category. The interval represents the last 60 seconds of processed requests. At the completion of the 60-second interval, all requests in the interval category are merged into the historical category.

Statistic	Description
Total Requests	The total number of incoming HTTP requests that have been processed since the Identity Server was started. Click **Graphs** to view the number of requests for a specific unit of time (1 hour, 1 day, 1 week, 1 month, 6 months, or 12 months). The Value axis displays the number of requests for the selected time period.
Currently Active Requests	The number of currently active incoming HTTP requests.
Oldest Active Request (Milliseconds)	The age of the oldest currently active incoming HTTP request.
Last Interval Maximum Request Duration (Milliseconds)	The age of the longest incoming HTTP request that was processed during the last 60-second interval.
Last Interval Mean Request Duration (Milliseconds)	The mean age of all incoming HTTP requests that were processed during the last 60-second interval.
Historical Maximum Request Duration (Milliseconds)	The age of the longest incoming HTTP request that was processed since the Identity Server was started.
Historical Mean Request Duration (Milliseconds)	The mean age of all incoming HTTP requests that were processed since the Identity Server was started.

Outgoing HTTP Requests

Outgoing HTTP requests are divided into three categories: active, interval, and historical. As soon as a request is complete, it is placed into the interval category. The interval represents the last 60 seconds of processed requests. At the completion of the 60-second interval, all requests in the interval category are merged into the historical category.

Statistic	Description
Total Requests	The total number of outgoing HTTP requests that have been processed since the Identity Server was started. Click **Graphs** to view the number of requests for a specific unit of time (1 hour, 1 day, 1 week, 1 month, 6 months, or 12 months). The Value axis displays the number of requests for the selected time period.
Currently Active Requests	The number of currently active outgoing HTTP requests.
Oldest Active Request (Milliseconds)	The age of the oldest currently active outgoing HTTP request.
Last Interval Maximum Request Duration (Milliseconds)	The age of the longest outgoing HTTP request that was processed during the last 60-second interval.
Last Interval Mean Request Duration (Milliseconds)	The mean age of all outgoing HTTP requests that were processed during the last 60-second interval.
Historical Maximum Request Duration (Milliseconds)	The age of the longest outgoing HTTP request that was processed, since the Identity Server was started.
Historical Mean Request Duration (Milliseconds)	The mean age of all outgoing HTTP requests that were processed, since the Identity Server was started.

Liberty

Statistic	Description
Liberty Federation	The number of Liberty protocol federations performed, since the Identity Server was started.
Liberty De-Federations	The number of Liberty protocol de-federations performed, since the Identity Server was started.
Liberty Register-Names	The number of Liberty protocol register names performed, since the Identity Server was started.

Clustering

An authoritative server is the cluster member that holds the authentication information for a given user session. For a request associated with a given session to be processed, it must be routed (“proxied”) to the authoritative cluster member. If an L4 switch causes a request to go to a non-authoritative cluster member, then that cluster member proxies that request to the authoritative cluster member.

When a request is received, a cluster member uses multiple means to determine which cluster member is the authoritative server for the request. It looks for a parameter on the query string of the URL indicating the authoritative server. It looks for an HTTP cookie indicating the authoritative server. If these do not exist, the cluster member examines the payload of the HTTP request to determine the authoritative server. Payload examinations result in immediate identification of the authoritative server or a user session ID or user identity ID that can be used to locate the authoritative server.

If a user session ID or user identity ID is found, the ID is broadcast to all cluster members asking which member is the authoritative server for the given ID. The authoritative server receives the broadcast message, determines that it indeed holds the given session or user, and responds accordingly.

The higher the number of proxied requests, the lower the performance of the entire system. Furthermore, the higher the number of payload examinations and ID broadcasts, the lower the performance of the entire system.

Statistic	Description
Currently Active Proxied Requests	The number of currently active proxied HTTP requests.
Total Proxied Requests	The total number of proxied requests that have been processed, since the Identity Server was started. These requests were sent to a non-authoritative (wrong) box.
Total Non-Proxied Requests	The total number of non-proxied requests that have been processed, since the Identity Server was started. These requests were sent to the authoritative (correct) box.
Authoritative Server Obtained from URL Parameter	The total number of authoritative servers identified by using the parameter from the URL query string, since the Identity Server was started.
Authoritative Server Obtained from Cookie	The total number of authoritative servers identified by using the HTTP cookie, since the Identity Server was started.
Payload Examinations	The total number of attempted payload examinations to identify the authoritative server, since the Identity Server was started.

Statistic	Description
Successful Payload Examinations	The total number of successful payload examinations to identify the authoritative server, since the Identity Server was started.
Identity ID Broadcasts	The total number of attempted Identity ID Broadcasts to identify the authoritative server, since the Identity Server was started.
Successful Identity ID Broadcasts	The total number of successful Identity ID Broadcasts to identify the authoritative server, since the Identity Server was started.
Session ID Broadcasts	The total number of attempted Session ID Broadcasts to identify the authoritative server, since the Identity Server was started.
Successful Session ID Broadcasts	The total number of successful Session ID Broadcasts to identify the authoritative server, since the Identity Server was started.

SP Brokering

Statistic	Description
Total Brokering Requests	The total number of brokering requests created after the Identity Server was started. This count is a sum of all connections created to all replicas of the configuration datastore and all user stores.
Total Brokering Requests Denied Due to Group Check	The total number of brokering authentication requests denied in a target service provider. The brokering group can either be the identity provider or target service provider but both does not belong to the same group.
Total Brokering Requests Denied Due to Role Deny	The total number of brokering authentication requests to a target service provider denied due to broker policy evaluation denying the role.
Total Brokering Requests Passed	The total number of brokering requests passed after the Identity Server was started.

Monitoring Access Gateway Cluster Statistics

You can view and configure general performance statistics for the servers and service providers assigned to the selected cluster.

Server Statistics

On the Cluster Statistics page, you can configure the list of server statistics to show the desired statistics for an Access Gateway cluster. See “Server Activity Statistics” on page 882 and “Server Benefits Statistics” on page 886 for the complete list of statistics for each server in an Access Gateway cluster.

Perform the following steps:

1. In the Administration Console, click **Devices** > **Access Gateways** > **[Name of Cluster]** > **Statistics** > **Server Statistics**.
2. Click **Configure**.
3. Select **Set default statistics under Selected Statistics** to replace the selected statistics with the default statistics.

The default statistics include CPU Utilization, Current Connections to Origin Server, Current Connections to Browsers, Total Bytes per Second, Requests per Second to Origin Server, Requests per Second from Browsers, and Total Requests per Second.

4 Select statistics from **Available Statistics** and move to **Selected Statistics**.

5 Click **OK**.

The Cluster Statistics page displays the summary of configured statistics for each individual member of the cluster.

To view additional statistical information about a specific Access Gateway, click the name of the Access Gateway in the **Server Name** column of the summary.

You can also view all the statistics for an individual server of the cluster. Click **View** in the **Statistics** column of the summary to see these additional statistics.

For more information, see "Monitoring Access Gateway Statistics" on page 882.

6 Click **Close**.

NOTE: In the cluster level statistics, you can view the list of servers, which are associated with that cluster. If a statistics is not applicable for a particular Access Gateway server, the value of the statistics is displayed as `Not Supported` for that server.

Service Provider Statistics

On the Cluster Statistics page, you can configure the list of service provider statistics to show the desired statistics for an Access Gateway cluster. See "Service Provider Activity Statistics" on page 887 for the complete list of statistics for each server in an Access Gateway cluster.

Perform the following steps:

1 In the Administration Console, click **Devices** > **Access Gateways** > **[Name of Cluster]** > **Statistics** > **Service Provider Statistics**.

2 Click **Configure**.

3 Select **Set default statistics under Selected Statistics** if you want to replace the selected statistics with the default statistics.

The default statistics include Free Memory, Provided Authentications, Cached Sessions, Logins In Last Interval, Currently Active Requests - Incoming HTTP Requests, Currently Active Requests - Outgoing HTTP Requests, and Currently Active Proxied Requests.

4 Select statistics from **Available Statistics** and move to **Selected Statistics**.

5 Click **OK**.

The Cluster Statistics page displays the summary of configured statistics for each individual member of the cluster.

To view additional statistical information about a specific Access Gateway, click the name of the Access Gateway in the **Server Name** column of the summary.

You can also view all the statistics for an individual server of the cluster. Click **View** in the **Statistics** column of the summary to see these additional statistics.

For more information, see "Service Provider Activity Statistics" on page 887.

6 Click **Close**.

Component Statistics Through REST APIs

You can programmatically access statistics of Access Gateways, Identity Servers, and ESP. This section includes the following topics:

- "Monitoring API for the Identity Server Statistics" on page 893
- "Monitoring API for the Access Gateway Statistics" on page 898

Monitoring API for the Identity Server Statistics

For programmatic access to the Identity Server statistics, you must enable the Representational State Transfer (REST) API.

To enable the REST API:

1. Place the `nidpmonitor.txt` file in to the `WEB-INF` directory of the Identity Server and ESP webapp.

   For Identity Server:

   `/opt/novell/nam/idp/webapps/nidp/WEB-INF/`

   For ESP:

   `/opt/novell/nam/mag/webapps/nesp/WEB-INF/`

2. Add the following line in `nidpmonitor.txt`:

   `urn:novell:nidp:monitor:anyaccess`

   After this line, you must add the IP addresses of the servers from which you will be making calls to the REST API. Example content of the `nidpmonitor.txt` file:

   urn:novell:nidp:monitor:anyaccess

   10.0.0.0

   172.16.0.0

3. Restart the Identity Server.

IMPORTANT: Frequent requests to get the statistics impact the system performance. It is recommended to keep a five minutes interval between every probe for the statistics.

Endpoints of the REST API

The Identity Server uses this REST endpoint: https://<DNS FQDN of NIDP>:<port>/nidp/app/monitor.

ESP uses this REST endpoint: https://<DNS FQDN of ESP>:<port>/nesp/app/monitor.

The endpoint takes the following three parameters:

Parameter	Value	Description
displayType	XML	This parameter specifies the output display type. Currently it supports only XML.
command	See Supported Commands and Their Outputs for details of the commands that support this parameter.	This specifies the monitored statistics that are to be displayed.
reset	This parameter can take only "True" as value. See Supported Commands and Their Outputs for details of the commands which support reset.	This specifies the monitored statistics that is to be reset.

Supported Commands and Their Outputs

The following list includes supported commands:

- httpInRequests
- inUrlTypes
- httpOutRequests
- ldapServerConfig
- ldapConnections
- ldapConnectionWaits
- ldapReplicaStats
- ldapPerfOverview
- ldapFailOverview
- authPerf

NOTE: When using the curl command, place the URL inside double quotes (""). Otherwise, the XML data does not render. For example, curl -k "https://<domain>:<port>/nidp/app/monitor?command=inUrlTypes&displayType=xml".

httpInRequests

This command supports reset. This command displays the monitored statistics of incoming HTTP requests to the Identity Server.

Example output:

```
<?xml version="1.0" encoding="UTF-8"?><InComingHTTPRequests> <ThreadIntervals>
<NamedValues> <NamedValue name="Total" value="61" /> <NamedValue name="Current
Requests" value="1" /> </NamedValues> <ActiveObjects abandoned="0"> <ActiveObject
name="ajp-bio-/127.0.0.1-9019-exec-23" age="3"> </ActiveObject> </ActiveObjects>
<Historical> <Spectrometer dataPoints="22" totalCount="60" maxDataPoints="500">
<max>145</max> <min>1</min> <mean>18</mean> </Spectrometer> </Historical> </
ThreadIntervals></InComingHTTPRequests>
```

inUrlTypes

This command supports reset. This command displays counts of the URL types and services that have been requested to the Identity Server.

Example output:

```
<UrlTypes> <NamedValues> <NamedValue name="CMD: /app/, monitor" value="15" />
<NamedValue name="CMD: /app/, ping" value="13" /> <NamedValue name="CMD: /idff,
soap" value="1" /> <NamedValue name="CMD: /idff, sso" value="4" /> <NamedValue
name="JSP: content.jsp" value="1" /> </NamedValues></UrlTypes>
```

httpOutRequests

This command supports reset. This command displays the monitored statistics of outgoing HTTP requests from the Identity Server.

Example output:

```
<?xml version="1.0" encoding="UTF-8"?><OutGoingHTTPRequests> <ThreadIntervals>
<NamedValues> <NamedValue name="Total" value="25" /> </NamedValues> <Historical>
<Spectrometer dataPoints="10" totalCount="25" maxDataPoints="500"> <max>51</max>
<min>2</min> <mean>12</mean> </Spectrometer> </Historical> </ThreadIntervals></
OutGoingHTTPRequests>
```

ldapServerConfig

This command does not support reset. This command displays the setup details of the Identity Server configuration store and the user store.

Example output:

```
<UserStoreManager id="MGf373f25e-5a95-484e-85fe-2d3f073e3c28">
<TrustConfigDataStore> <UserStore id="USef25d609-7577-4bab-a705-f00b5406f2cc"
systemId="cn=SCC7u0ouw,cn=cluster,cn=nids,ou=accessManagerContainer,o=novell"
displayName="" directoryName="Novell eDirectory"
adminUserName="ou=nidsUser,ou=UsersContainer,ou=Partition,ou=PartitionsContainer,o
u=VCDN_Root,ou=accessManagerContainer,o=novell" idleTimeout="10000"
bindTimeout="0" allowRebind="true" maxWaitReservations="-1"> <Replicas> <Replica
id="0c498978-2d16-4b25-ae41-484fca62fc36"
systemId="PseudoXMLBasedUserStoreReplicaDN0" displayName="Replica 1" host="ldaps:/
/ 10.0.0.0" port="636" maxConnections="5" doSSL="true"> <ConnectionPool
id="PL8928e311-6a84-494a-b61a-5ff43005dd6f:0c498978-2d16-4b25-ae41-484fca62fc36"
adminUserName="ou=nidsUser,ou=UsersContainer,ou=Partition,ou=PartitionsContainer,o
u=VCDN_Root,ou=accessManagerContainer,o=novell" maxConnections="5" skipCount="10"
waitResTimeout="60000" waitResSleep="20" waitResSleepIterCount="3000" load="0">
<AdminConnections> <Connection id="0adff495-9321-485c-b156-66deceeefa84"
type="admin" checkedOut="false" IdleAge="5985087" /> </AdminConnections> </
ConnectionPool> </Replica> </Replicas> </UserStore> </TrustConfigDataStore>
<UserStores> <UserStore id="USc15e7906-d4a9-41c3-8438-cd10fb6c7a89"
systemId="cn=USmkp9m,cn=Alrre4,cn=SCC7u0ouw,cn=cluster,cn=nids,ou=accessManagerCon
tainer,o=novell" displayName="SingleBoxUserStore" directoryName="Novell
eDirectory" adminUserName="cn=admin,o=novell" idleTimeout="10000" bindTimeout="0"
allowRebind="true" maxWaitReservations="-1"> <SearchContexts> <SearchContext
order="0" scope="1" context="o=novell" /> </SearchContexts> <Replicas> <Replica
id="0a307605-8946-4455-8080-f1819562481d"
systemId="cn=USRlxnx69,cn=USmkp9m,cn=Alrre4,cn=SCC7u0ouw,cn=cluster,cn=nids,ou=acc
essManagerContainer,o=novell" displayName="SingleBoxUserStoreReplica"
host="ldaps:// 10.0.0.0" port="636" maxConnections="20" doSSL="true">
```

```
<ConnectionPool id="PLce0653bc-488d-4e7c-81a5-08e935d83c82:0a307605-8946-4455-
8080-f1819562481d" adminUserName="cn=admin,o=novell" maxConnections="20"
skipCount="10" waitResTimeout="60000" waitResSleep="20"
waitResSleepIterCount="3000" load="0"> <AdminConnections> <Connection
id="b1c0a413-2c36-4b64-831c-b0849421c7a0" type="admin" checkedOut="false"
IdleAge="259357" /> </AdminConnections> </ConnectionPool> </Replica> </Replicas>
</UserStore> </UserStores></UserStoreManager>
```

ldapConnections

This command does not support reset. This command displays counts of the Identity Server LDAP connection.

Example output:

```
<LdapConnections> <TotalAdded admin="25" user="1" /> <TotalRemoved admin="23"
user="1" /> <CurrentValidInUse admin="0" user="0" /> <CurrentValidOutOfUse
admin="2" user="0" /> <CurrentInvalidEstd admin="0" user="0" />
<CurrentInvalidNonEstd admin="0" user="0" /> <TotalForceCloseSuccess admin="23"
user="1" /> <TotalForceCloseError admin="0" user="0" /> <TotalForceCloseNonEstd
admin="0" user="0" /></LdapConnections>
```

ldapConnectionWaits

This command supports reset. This command displays statistics of the Identity Server LDAP connection wait time.

Example output:

```
<LDAPConnectionWaits></LDAPConnectionWaits>
```

ldapReplicaStats

This command does not support reset. This command displays statistics of the Identity Server LDAP replica.

Example output:

```
<LdapReplicaStatsCollection> <TrustConfigDataStoreStats> <LdapReplicaStats
displayName="Replica 1" host="ldaps:// 10.0.0.0 " inRestart="false" load="0">
<ExistingAdminConnectionReservation admin="97" /> <NewConnections admin="2"
user="0" /> <Rebinds user="0" /> <InvalidRebinds user="0" /> <Waits admin="0"
user="0" /> <WaitExpired admin="0" user="0" /> <WaitSkipped admin="0" user="0" />
<WaitHitMaxSkipped admin="0" user="0" /> </LdapReplicaStats> </
TrustConfigDataStoreStats> <LdapReplicaStats
displayName="SingleBoxUserStoreReplica" host="ldaps://10.0.0.0" inRestart="false"
load="0"> <ExistingAdminConnectionReservation admin="86" /> <NewConnections
admin="28" user="1" /> <Rebinds user="0" /> <InvalidRebinds user="0" /> <Waits
admin="0" user="0" /> <WaitExpired admin="0" user="0" /> <WaitSkipped admin="0"
user="0" /> <WaitHitMaxSkipped admin="0" user="0" /> </LdapReplicaStats></
LdapReplicaStatsCollection>
```

ldapPerfOverview

This command does not support reset. This command displays performance statistics of the Identity Server LDAP replica.

Example output:

<?xml version="1.0" encoding="UTF-8"?><LdapReplicaPerfCollection>
<TrustConfigDataStorePerf> <LdapReplicaPerf displayName="Replica 1"
inRestart="false" load="0" host="ldaps://10.0.0.0"> <AllOpsDuration> <Interval>
<Spectrometer dataPoints="5" totalCount="6" maxDataPoints="300"> <max>46</max>
<min>1</min> <mean>16</mean> </Spectrometer> </Interval> <Historical>
<Spectrometer dataPoints="11" totalCount="100" maxDataPoints="500"> <max>93</max>
<min>1</min> <mean>3</mean> </Spectrometer> </Historical> </AllOpsDuration>
<CreateConnDuration> <Interval> <Spectrometer dataPoints="2" totalCount="2"
maxDataPoints="300"> <max>46</max> <min>44</min> <mean>45</mean> </Spectrometer>
</Interval> <Historical> <Spectrometer dataPoints="1" totalCount="1"
maxDataPoints="500"> <max>93</max> <min>93</min> <mean>93</mean> </Spectrometer>
</Historical> </CreateConnDuration> <CloseConnDuration> <Interval> <Spectrometer
dataPoints="1" totalCount="2" maxDataPoints="300"> <max>1</max> <min>1</min>
<mean>1</mean> </Spectrometer> </Interval> </CloseConnDuration> <SearchDuration>
<Interval> <Spectrometer dataPoints="2" totalCount="2" maxDataPoints="300">
<max>3</max> <min>2</min> <mean>2</mean> </Spectrometer> </Interval> <Historical>
<Spectrometer dataPoints="8" totalCount="95" maxDataPoints="500"> <max>11</max>
<min>1</min> <mean>2</mean> </Spectrometer> </Historical> </SearchDuration>
<GetDuration> <Historical> <Spectrometer dataPoints="4" totalCount="4"
maxDataPoints="500"> <max>10</max> <min>1</min> <mean>6</mean> </Spectrometer> </
Historical> </GetDuration> <ModifyDuration></ModifyDuration> <CreateObjDuration></
CreateObjDuration> <DeleteObjDuration></DeleteObjDuration> <ExtDuration></
ExtDuration> <RebindDuration></RebindDuration> </LdapReplicaPerf> </
TrustConfigDataStorePerf> <LdapReplicaPerf displayName="SingleBoxUserStoreReplica"
inRestart="false" load="0" host="ldaps://10.0.0.0"> <AllOpsDuration> <Interval>
<Spectrometer dataPoints="5" totalCount="19" maxDataPoints="300"> <max>46</max>
<min>1</min> <mean>13</mean> </Spectrometer> </Interval> <Historical>
<Spectrometer dataPoints="5" totalCount="9" maxDataPoints="500"> <max>43</max>
<min>0</min> <mean>5</mean> </Spectrometer> </Historical> </AllOpsDuration>
<CreateConnDuration> <Interval> <Spectrometer dataPoints="2" totalCount="5"
maxDataPoints="300"> <max>46</max> <min>45</min> <mean>45</mean> </Spectrometer>
</Interval> <Historical> <Spectrometer dataPoints="1" totalCount="1"
maxDataPoints="500"> <max>43</max> <min>43</min> <mean>43</mean> </Spectrometer>
</Historical> </CreateConnDuration> <CloseConnDuration> <Interval> <Spectrometer
dataPoints="1" totalCount="5" maxDataPoints="300"> <max>1</max> <min>1</min>
<mean>1</mean> </Spectrometer> </Interval> </CloseConnDuration> <SearchDuration>
<Interval> <Spectrometer dataPoints="2" totalCount="3" maxDataPoints="300">
<max>2</max> <min>1</min> <mean>1</mean> </Spectrometer> </Interval> </
SearchDuration> <GetDuration> <Interval> <Spectrometer dataPoints="2"
totalCount="3" maxDataPoints="300"> <max>2</max> <min>1</min> <mean>1</mean> </
Spectrometer> </Interval> <Historical> <Spectrometer dataPoints="2" totalCount="4"
maxDataPoints="500"> <max>2</max> <min>1</min> <mean>1</mean> </Spectrometer> </
Historical> </GetDuration> <ModifyDuration></ModifyDuration> <CreateObjDuration></
CreateObjDuration> <DeleteObjDuration></DeleteObjDuration> <ExtDuration>
<Interval> <Spectrometer dataPoints="2" totalCount="2" maxDataPoints="300">
<max>2</max> <min>1</min> <mean>1</mean> </Spectrometer> </Interval> <Historical>
<Spectrometer dataPoints="3" totalCount="4" maxDataPoints="500"> <max>3</max>
<min>0</min> <mean>1</mean> </Spectrometer> </Historical> </ExtDuration>
<RebindDuration> <Interval> <Spectrometer dataPoints="1" totalCount="1"
maxDataPoints="300"> <max>3</max> <min>3</min> <mean>3</mean> </Spectrometer> </
Interval> </RebindDuration> </LdapReplicaPerf></LdapReplicaPerfCollection>

ldapFailOverview

This command does not support reset. This command displays statistics of the Identity Server LDAP replica failure.

Example output:

```
<?xml version="1.0" encoding="UTF-8"?><LdapReplicaFailureCollection>
<TrustConfigDataStoreFailure> <LdapReplicaFailurePerf displayName="Replica 1"
inRestart="false" load="0" host="ldaps://10.0.0.0"> <AllOpsDuration> <Historical>
<Spectrometer dataPoints="2" totalCount="3" maxDataPoints="500"> <max>2</max>
<min>1</min> <mean>1</mean> </Spectrometer> </Historical> </AllOpsDuration>
<CreateConnDuration></CreateConnDuration> <CloseConnDuration></CloseConnDuration>
<SearchDuration></SearchDuration> <GetDuration> <Historical> <Spectrometer
dataPoints="2" totalCount="3" maxDataPoints="500"> <max>2</max> <min>1</min>
<mean>1</mean> </Spectrometer> </Historical> </GetDuration> <ModifyDuration></
ModifyDuration> <CreateObjDuration></CreateObjDuration> <DeleteObjDuration></
DeleteObjDuration> <ExtDuration></ExtDuration> <RebindDuration></RebindDuration>
</LdapReplicaFailurePerf> </TrustConfigDataStoreFailure> <LdapReplicaFailurePerf
displayName="SingleBoxUserStoreReplica" inRestart="false" load="0" host="ldaps://
10.0.0.0"> <AllOpsDuration> <Interval> <Spectrometer dataPoints="2" totalCount="2"
maxDataPoints="300"> <max>3054</max> <min>3051</min> <mean>3052</mean> </
Spectrometer> </Interval> </AllOpsDuration> <CreateConnDuration> <Interval>
<Spectrometer dataPoints="2" totalCount="2" maxDataPoints="300"> <max>3054</max>
<min>3051</min> <mean>3052</mean> </Spectrometer> </Interval> </
CreateConnDuration> <CloseConnDuration></CloseConnDuration> <SearchDuration></
SearchDuration> <GetDuration></GetDuration> <ModifyDuration></ModifyDuration>
<CreateObjDuration></CreateObjDuration> <DeleteObjDuration></DeleteObjDuration>
<ExtDuration></ExtDuration> <RebindDuration></RebindDuration> </
LdapReplicaFailurePerf></LdapReplicaFailureCollection>
```

authPerf

This command does not support reset. This command displays performance statistics of the Identity Server local authentication.

Example output:

```
<?xml version="1.0" encoding="UTF-8"?><AuthenticationPerformance> <NamedValues>
<NamedValue name="Provided Authentications" value="2" /> <NamedValue
name="Consumed Authentications" value="3" /> <NamedValue name="Consumed
Authentications Failures" value="6" /> <NamedValue name="Historical PEAK Logins"
value="1" /> <NamedValue name="Logouts" value="2" /> </NamedValues>
<LocalAuthDuration historicalMean="106" intervalMean="105"> <ContractStats
name="Name/Password - Form"> <Historical> <Spectrometer dataPoints="1"
totalCount="1" maxDataPoints="500"> <max>100</max> <min>100</min> <mean>100</mean>
</Spectrometer> </Historical> </ContractStats> <ContractStats
name="MyTwoContracts"> <Interval> <Spectrometer dataPoints="1" totalCount="1"
maxDataPoints="300"> <max>105</max> <min>105</min> <mean>105</mean> </
Spectrometer> </Interval> <Historical> <Spectrometer dataPoints="1" totalCount="1"
maxDataPoints="500"> <max>113</max> <min>113</min> <mean>113</mean> </
Spectrometer> </Historical> </ContractStats> </LocalAuthDuration> </
AuthenticationPerformance>
```

Monitoring API for the Access Gateway Statistics

For programmatic access to the Access Gateway statistics, you require to enable the global advanced option NAGStatsClientIPWhitelist. This option takes a list of IP addresses of servers that can access the Access Gateway statistics.

To access the statistics, run the HTTP GET command on the resource: https://<mag-host-name>/mag-stats.

NOTE: Frequent requests to get the statistics impact the system' performance. It is recommended to keep a five minutes interval between every probe for the statistics.

To enable this option:

1. In the Administration Console, select **Devices** > **Access Gateway Servers** > **Edit** > **Advanced Options**.
2. Add this line: `NAGStatsClientIPWhitelist <ip1> <ip2>`.
3. Replace <ip1> and <ip2> with the IP addresses of the servers from which you want to access the statistics.
4. Click **OK**.

This request displays the following:

- Https related statistics
  - Requests received
  - Active requests
- Server related statistics
  - Product start time
  - Product up time
  - Product CPU utilization
  - Disk swap (KB)
  - Disk swap used (KB)
  - Memory total (KB)
- Cache statistics

  NOTE: Cache statistics are 0 because they are not implemented currently in the server side.

  - Cache stats (KB)
  - Cache stats utilization percentage
  - Cache hit ratio since last reset
  - Cache stats object count
- Summary Statistics Byte
  - Total bytes sent to the origin server
  - Total bytes read from the Web server
  - Total bytes sent to the browsers
  - Total bytes received from the browsers
  - Bytes per sec read from the Web server
  - Bytes per sec sent to the browsers
- Summary Statistics Benefits
  - Total bytes saved
  - Total bytes saved per second

Example output:

<?xml version="1.0" encoding="UTF-8"?><MAGStatistics><httpStats> <NamedValues>
<NamedValue name="RequestsReceived" value="0" /> <NamedValue name="ActiveRequests"
value="1" /> </NamedValues></httpStats><boxStats> <NamedValues> <NamedValue
name="ProductStartTime" value="Fri, 27 Jul 2012 11:01:11 GMT"/> <NamedValue
name="ProductUpTime" value="0:0:0:26" /> <NamedValue name="ProductCPUUtilization"
value="-294" /> <NamedValue name="DiskSwapKb" value="4088532" /> <NamedValue
name="DiskSwapUsedKb" value="0" /> <NamedValue name="MemoryTotalKb" value="7835" /
> </NamedValues></boxStats><cacheStats> <NamedValues> <NamedValue
name="cacheStatsKb" value="0" /> <NamedValue name="cacheStatsUtilPercentage"
value="0" /> <NamedValue name="cacheHitRatioSinceReset" value="0" /> <NamedValue
name="cacheStatsObjectCount" value="0" /> </NamedValues></
cacheStats<summaryStatsByte> <NamedValues> <NamedValue
name="TotalBytesSentToOriginServer" value="0" /> <NamedValue
name="TotalBytesReadFromWS" value="0" /> <NamedValue
name="TotalBytesSentToBrowsers" value="0" /> <NamedValue
name="TotalBytesReceivedFromBrowsers" value="0" /> <NamedValue
name="BytesPsecReadFromWS" value="0" /> <NamedValue name="BytesPsecSentToBrowsers"
value="0" /> </NamedValues></summaryStatsByte><summaryStatsBenefits> <NamedValues>
<NamedValue name="TotalBytesSaved" value="0" /> <NamedValue
name="TotalBytesSavedPerSecond" value="0" /> </NamedValues></
summaryStatsBenefits><summaryStatsRequests> <NamedValues> <NamedValue
name="TotalRequestsPSecBrowsers" value="0" /> <NamedValue
name="PeakTotalRequestsPSecBrowsers" value="1" /> <NamedValue
name="TotalRequestsPSecOriginServer" value="0" /> <NamedValue
name="PeakTotalRequestsPSecOriginServer" value="0" /> <NamedValue
name="CurrentTotalRequestsToOriginServer" value="0" /> <NamedValue
name="CurrentTotalRequestsReceivedFromBrowser" value="1" /> <NamedValue
name="FailedRequestsToWS" value="0" /> <NamedValue name="CumulativeRequestsToWS"
value="0" /> </NamedValues></summaryStatsRequests><summaryStatsConnections>
<NamedValues> <NamedValue name="CurrentConnectionsBrowser" value="10" />
<NamedValue name="CurrentConnectionsBackend" value="0" /> <NamedValue
name="TotalConnectionsBrowser" value="28" /> <NamedValue
name="TotalConnectionsBackend" value="0" /> <NamedValue
name="PeakConnectionsBrowser" value="6" /> <NamedValue
name="PeakConnectionsBackend" value="0" /> <NamedValue
name="FailedConnectionsBackend" value="0" /> </NamedValues></
summaryStatsConnections></MAGStatistics>

20 Monitoring Server Health

You can monitor all components hosted by a server and quickly isolate and correct server issues. The system displays statuses (green, yellow, white, or red) for Access Manager components. You can access the health information at the following places:

- **Dashboard**

  The Dashboard page shows the heath status at the component-level.

- **Auditing > Device Health**

  The Device Health page shows the health status for all devices in one list.

- **Devices > [Component]**

  The Servers page for each component provides a health status for each device.

- "Health States" on page 901
- "Monitoring Health by Using the Hardware IP Address" on page 902
- "Monitoring Health of Identity Servers" on page 902
- "Monitoring the Health of Access Gateways" on page 904

Health States

The Health page displays the current status of the server. The following states are possible:

Icon	Description
	A green status indicates that the server has not detected any problems.
	A green status with a yellow diamond indicates that the server has not detected any problems but the configuration is not' completely up-to-date because commands are pending.
	A green status with a red x indicates that the server has not detected any problems but that the configuration might not be what you want because one or more commands have failed.
	A red status with a bar indicates that the server has been stopped.
	A white status with disconnected bars indicates that the server is not communicating with the Administration Console.
	A yellow status indicates that the server might be functioning sub-optimally because of configuration discrepancies.
	A yellow status with a question mark indicates that the server has not been configured.
	A red status with an x indicates that the server configuration might be incomplete or wrong, that a dependent service in not running or functional, or that the server is having a runtime problem.

Monitoring Health by Using the Hardware IP Address

The Hardware IP Address page allows you to view the devices and agents managed through the selected IP address. You can monitor all of the devices hosted by a server and quickly isolate and correct server issues. The system displays statuses (green, yellow, white, or red) for the Access Manager devices.

1. In the Administration Console, click **Auditing > Device Health**.
2. To view information about the health of each installed device, click an IP address.
3. Select one of the following actions:
   - To return to the Device Health page, click **Close**.
   - To edit the details of a device, click the server name.
   - To view health details, click the **Health** icon.
   - To view the alerts, click the alerts link.
   - To view device statistics, click the statistics link.
   - To view or configure audit events for the device, click the **Edit Events** link.

Monitoring Health of Identity Servers

This section discusses the following topics:

- "Monitoring the Health of an Identity Server" on page 902
- "Monitoring the Health of a Cluster" on page 904

Monitoring the Health of an Identity Server

To view detailed health status information for an Identity Server:

1. In the Administration Console, click **Devices > Identity Servers** > **[Name of Server]** > **Health**.

   The status icon is followed by a description that explains the significance of the current state. For more information about the icons, see "Health States" on page 901.
2. To ensure that the information is current, select one of the following:
   - Click **Refresh** to refresh the page with the latest health available from the Administration Console.
   - Click **Update from Server** to send a request to the Identity Server to update its status information. This can take a few minutes.

3 Examine the **Services Detail** section that displays the status of each service. For an Identity Server, this includes information such as the following:

Status Category	If not healthy
Status: Indicates whether the Identity Server is online and operational.	Verify whether the Identity Server has been stopped or is not configured. Also verify that network problems are not interfering with communications between the Identity Server and the Administration Console.
Services: Indicates the general health of all configured services.	If one service is unhealthy, this category reflects that status. See the particular service that also displays an unhealthy status.
Identity Server Configuration: Indicates the status of the configuration.	Configure the Identity Server or assign the server to a configuration.See,"Setting up User Stores for Identity Server Configuration" on page 52
Configuration Datastore: Indicates the status of the installed configuration datastore.	You might need to restart Tomcat or reinstall the Administration Console.
User Datastores: Indicates whether the Identity Server can communicate with the user stores, authenticate as the admin user, and find the search context.	Ensure that the user store is operating and configured correctly. You might need to import the SSL certificate for communication with the Identity Server. See "Configuring Identity User Stores" on page 272.
Signing, Encryption and SSL Connector Keys: Indicates whether these keystores contain valid a key.	Click **Identity Servers** > **Edit** > **Security** and replace any missing or expired keys.
System Incoming and Outgoing HTTP Requests: Appears when throughput is slow. This health check monitors incoming HTTP requests, outgoing HTTP requests on the SOAP back channel, and HTTP proxy requests to cluster members. If one or more requests remain in the queue for over 2 minutes, this health check appears.	Verify that all members of the cluster have sufficient bandwidth to handle requests. If a cluster member is going down, the problem resolves itself as other members of the cluster are informed that the member is down. If a cluster member is slow because it does not' have enough physical resources (speed or memory) to handle the load, upgrade the hardware.
SSL Communication: Indicates whether SSL communication is operating correctly. This health check appears only when the SSL communication check fails.	Check SSL connectivity. Check for expired SSL certificates.
Audit Logging Server: Indicates whether the audit agent is functioning and able to log events to the auditing server. Auditing must be enabled on the Identity Server to activate this health check (click **Devices** > **Identity Servers** > **Edit** > **Logging**).	Check the network connection between the Identity Server and the auditing server. See "Troubleshooting Novell Audit" (http://www.novell.com/documentation/novellaudit20/novellaudit20/data/al0lh30.html).

4 Click **Close**.

Monitoring the Health of a Cluster

The health page displays the current health of the cluster.

1. In the Administration Console, click **Devices > Identity Servers** > **[Name of Cluster]** > **Health**.

   The status icon is followed by a description that explains the significance of the current state. For more information about the icons, see “Health States” on page 901.
2. To ensure that the information is current, click **Refresh** to refresh the page with the latest health available from the Administration Console.
3. To view health details about a specific member of the cluster, click the server’s health icon.

Monitoring the Health of Access Gateways

This section discusses the following topics:

- “Monitoring the Health of an Access Gateway” on page 904
- “Monitoring the Health of an Access Gateway Cluster” on page 906

Monitoring the Health of an Access Gateway

To view detailed health status information of an Access Gateway:

1. In the Administration Console, click **Devices > Access Gateways** > **[Name of Server]** > **Health**.

   The status icon is followed by a description that explains the significance of the current state. For more information about these icons, see “Health States” on page 901.
2. To ensure that the information is current, select one of the following:
   - Click **Refresh** to refresh the page with the latest health available from the Administration Console.
   - Click **Update from Server** to send a request to the Access Gateway to update its status information. If you have made changes that affect the health of the Access Gateway, select this option. Otherwise, it can take up to five minutes for the health status to change.
3. Examine the **Services Detail** section that displays the status of each service. For an Access Gateway, this includes information such as the following:
   - “Service Categories of the Access Gateway Service” on page 904
4. Click **Close**.

Service Categories of the Access Gateway Service

Service Category	If Not Healthy
Reverse Proxy - *<Proxy Service Name>*: Indicates the general health of all configured proxy services. A separate row is created for each proxy service.	Check the health of the Web server.

Service Category	If Not Healthy
AGM - Configuration: Indicates whether all configuration changes have been applied.	Do the following: ◆ To re-push the current configuration, click **Troubleshooting**, select the gateway from the list of the **Current Access Gateway Configurations**, then click **Re-push Current Configuration**. ◆ To revert to last applied configuration, click **Devices** > **Access Gateways** > **Edit**, then click **Revert**. If these options do not fix the problem, view the Apache `error.log` file to discover the cause. The file is located in the following directory:
TCP Listener - *<IP Address:Port>*: Indicates whether the Access Gateway Service is listening on the specified port. A separate row is created for each port the Gateway Service is configured to listen on.	Restart the Apache service.
ApacheGateway.log: Appears when the Access Gateway Service is not healthy. It displays the latest error from the Apache `error.log` file.	For more information about the problem, view the `error.log` file in the following directory:
Embedded Service Provider Configuration: Indicates whether the Access Gateway has been configured to trust an Identity Server and whether that configuration has been applied. At least one Identity Server must be configured and set up as a trusted authentication source for the Access Gateway. A green status indicates that a configuration has been applied; it does not indicate that it is a functioning configuration.	See "Managing Reverse Proxies and Authentication" on page 94 for information about assigning an Identity Server configuration to the Access Gateway.
Configuration Datastore: Indicates whether the configuration datastore is functioning correctly.	Restore the configuration datastore. See "Repairing the Configuration Datastore" on page 956.
Clustering: Indicates whether all the cluster members are active and processing requests.	Restart the cluster members that are not active or remove them from the cluster.
Signing, Encryption and SSL Connector Keys: Indicates whether these keystores contain a valid key.	Click **Access Gateways** > **Edit** > **Service Provider Certificates** and replace any missing or expired keys.
System Incoming and Outgoing HTTP Requests: Appears when throughput is slow. This health check monitors incoming HTTP requests, outgoing HTTP requests on the SOAP back channel, and HTTP proxy requests to cluster members. If one or more requests remain in the queue for over 2 minutes, this health check appears.	Verify that all members of the cluster have sufficient bandwidth to handle requests. If a cluster member is going down, the problem resolves itself as other members of the cluster are informed that the member is down. If a cluster member is slow because it does not' have enough physical resources (speed or memory) to handle the load, upgrade the hardware.
TCP Listener(s): Indicates whether the listening port for the Embedded Service Provider is healthy.	Restart the Access Gateway.

Service Category	If Not Healthy
Embedded Service Provider's Trusted Identity Provider: Indicates whether the configuration that the Access Gateway trusts has been configured to contain at least one Identity Server.	Modify the Identity Server configuration and add an Identity Server. Configure the Access Gateway to trust an Identity Server configuration. See "Creating a Proxy Service" on page 95.
Audit Logging Server: Indicates whether the audit agent is functioning and able to log events to the auditing server. Auditing must be enabled on the Identity Server to activate this health check (click **Devices** > **Identity Servers** > **Edit** > **Logging**).	Check the network connection between the Identity Server and the auditing server. See "Troubleshooting Novell Audit" (http://www.novell.com/documentation/novellaudit20/novellaudit20/data/al0lh30.html).

Monitoring the Health of an Access Gateway Cluster

The **Health** icon on the cluster row displays the status of the least healthy member of the cluster. For information about the meaning of health icons, see "Health States" on page 901.

To view details about the status of the cluster:

1. In the Administration Console, click **Devices** > **Access Gateways**.
2. On the cluster row, click the **Health** icon.
3. To ensure that the information is current, click **Refresh**.
4. To view specific information about the status of an Access Gateway, click the **Health** icon in the Access Gateway row.

21 Monitoring Component Command Status

Commands are issued to a component when you make configuration changes and when you select an action such as stopping or starting that component. The command status only displays the commands of certificates that are associated with a device.

Certain commands, such as start and stop, retry up to 10 times before they fail. The first few retries are spaced a few minutes apart, then they move to 10-minute intervals. These commands can take over an hour to result in a failure. As long as the command is in the retry cycle, the command has a status of pending.

- If you do not want to wait for the cycle to complete, you need to manually delete the command.
- If you enter the same command and it succeeds before the first command has completed its retry cycle, the first command always stays in the pending state. You need to manually delete the command.

The Command Status page lists scheduled events and the current status of each event. A new command appears in the list each time you change a configuration. The commands remain listed until you delete them.

This section discusses the following topics:

- "Viewing the Command Status of the Identity Server" on page 907
- "Viewing the Command Status of the Access Gateway" on page 908
- "Reviewing the Command Status for Certificates" on page 910

Viewing the Command Status of the Identity Server

This section describes the following tasks related to commands:

- "Viewing the Status of Current Commands" on page 907
- "Viewing Detailed Command Information" on page 908

Viewing the Status of Current Commands

1. In the Administration Console, click **Devices > Identity Servers**.
2. Click the **Command Status** link for the server.
3. To delete a command, select it and click **Delete**.
4. Click **Refresh** to refresh the display.

   The following table describes the columns on the Command Status page:

Column Name	Description
Name	Lists the Identity Server name.

Column Name	Description
Status	Lists the status of each server.
Type	Displays type of command issued to the server.
Admin	Displays the credentials of the administrator who performed the command.
Date & Time	The date and time that the command was issued. Date and time entries are specified in the local time.

Viewing Detailed Command Information

To view information about an individual command:

1. In Administration Console, click **Devices > Identity Servers** > *[Name of Server]* > **Command Status.**
2. Click the name of a command to get detailed information. The following information is displayed:

   Name: The Identity Server name.

   Type: The type of command issued to the server.

   Admin: The distinguished name of the admin user who performed the command.

   Status: The status of the server command.

   Last Executed On: The date and time that the command was executed.
3. To determine if any problems occurred, view the **Command Execution Details** section.

   For a command that fails because the Administration Console cannot communicate with the Identity Server, the page displays the following additional fields:

   Number of Tries: Specifies the number of times the command was executed.

   Command Try Log: Lists each try and the results.
4. Select one of the following actions:
   - **Delete:** To delete a command, click **Delete**. Click **OK** in the confirmation dialog box.
   - **Refresh:** To update the current cache of recently executed commands, click **Refresh**.
5. Click **Close** to return to the Command Status page.

Viewing the Command Status of the Access Gateway

This section describes the following tasks related to commands:

- "Viewing the Status of Current Commands" on page 908
- "Viewing Detailed Command Information" on page 909

Viewing the Status of Current Commands

1. Click **Devices** > **Access Gateways** > **[Name of Server]** > **Command Status**.

Column Name	Description
Name	Specifies name of the command. Click the link to view additional details about the command. For more information, see "Viewing Detailed Command Information" on page 909.
Status	Specifies status of the command. Some of the possible states of the command include Pending, Incomplete, Executing, and Succeeded.
Type	Specifies the type of command.
Admin	Specifies if the system or a user issued the command. If a user issued the command, the DN of the user is displayed.
Date & Time	Specifies the local date and time the command was issued.

2 Select one of the following actions:

- To view information about a particular command, click the name of a command.
- To delete a command from the list, select the command, then click **Delete**.
- To refresh the status of the listed commands, click **Refresh**.

3 Click **Close**.

Viewing Detailed Command Information

1 In Administration Console, click **Devices > Access Gateways** > *[Name of Server]* > **Command Status.**

2 Click the name of a command to get detailed information.

The following command information is listed:

Name: Specifies the display name that has been given to the command.

Type: Specifies the type of command.

Admin: Specifies whether the system or a user issued the command. If a user issued the command, the field contains the DN of the user.

Status: Specifies the status of the command, and includes such states as **Pending**, **Incomplete**, **Executing**, and **Succeeded**.

Last Executed On: Specifies when the command was issued. The date and time are displayed in local time. If the command failed, additional information is available.

For a command that the Administration Console can successfully send to the Access Gateway, the page displays a **Command Execution Details** section with the name of the command and the command results.

For a command that fails because the Administration Console cannot communicate with the Access Gateway, the page displays the following additional fields:

Number of Tries: Specifies the number of times the command was executed.

Command Try Log: Lists each try and the results.

3 You can delete a command or update the current cache of recently executed commands.

4 Click **Close** to return to the Command Status page.

Reviewing the Command Status for Certificates

The command status displays only commands of certificates that are associated with a device.You can view the status of the commands that have been sent to the certificate server for execution.

1 In the Administration Console, click **Security** > **Certificates**, then click **Command Status**.

2 Use the following options to review or change a server's certificate command status:

- **Delete:** To delete a command, select the check box for the command, then click **Delete**. The selected command is cleared.
- **Refresh:** Click **Refresh** to update the current cache of recently executed commands.
- **Name:** Click this box to select all the commands in the list, then click **Refresh** or **Delete**.

The following table describes the features on this page:

Column Name	Description
Name	Contains the display name of the command. Click the link to view additional details about the command.
Status	Specifies the status of the command. Some of the possible states of the command include `Pending`, `Incomplete`, `Executing`, and `Succeeded`.
Type	Specifies the type of server, such as Identity Server or Access Gateway.
Commands	Specifies the command given, such as `Import certificate`, or `Import trusted root`.
Admin	Specifies if the system or a user issued the command. If a user issued the command, the DN of the user is displayed.
Date & Time	Specifies the local date and time the command was issued.

3 To review command information, click a link under the **Name** column.

This page displays status information about the command and allows you to perform the following tasks:

Refresh: Select this option to refresh the data for this command.

Delete: Select this option to clear this command.

The following command information is listed:

Name: Specifies the display name that has been given to the command.

Type: Specifies the type of command.

Admin: Specifies whether the system or a user issued the command. If a user issued the command, the field contains the DN of the user.

Status: Specifies the status of the command, and includes such states as **Pending**, **Incomplete**, **Executing**, and **Succeeded**.

Last Executed On: Specifies when the command was issued. The date and time are displayed in local time. If the command failed, additional information is available.

For a command that the Administration Console can successfully process, the page displays a **Command Execution Details** section with the name of the command and the command results.

4 Click **Close**.

22 Monitoring Alerts

An alert is generated whenever the system detects a condition that prevents it from performing normal system services. Access Manager components have been programmed to issue alerts to various types of systems (such as a Novell Audit server, a NetIQ Sentinel server, or a Syslog server) so that the administrator can be informed when significant changes occur that modify Access Manager performance. Alerts can also be configured so that the administrator is informed when significant changes occur.

This section discusses the following topics:

- "Monitoring Identity Server Alerts" on page 911
- "Monitoring Access Gateway Alerts" on page 911

Monitoring Identity Server Alerts

1. In the Administration Console, click **Devices** > **Identity Servers** > **[Name of Server]** > **Alerts** tab.
2. To delete an alert from the list, select the check box for the alert, then click **Acknowledge Alert(s)**. To remove all alerts from the list, click the **Severity** check box, then click **Acknowledge Alert(s)**.
3. Click **Close**.
4. (Optional) To verify that the problem has been solved, click **Identity Servers** > **[Name of Server]** > **Health** > **Update from Server**.

Monitoring Access Gateway Alerts

This section discusses the following topics:.

- "Viewing Access Gateway Alerts" on page 911
- "Viewing Access Gateway Cluster Alerts" on page 912
- "Managing Access Gateway Alert Profiles" on page 912
- "Configuring an Alert Profile" on page 912
- "SNMP Profile" on page 914
- "Configuring a Log Profile" on page 914
- "Configuring an E-Mail Profile" on page 914
- "Configuring a Syslog Profile" on page 915

Viewing Access Gateway Alerts

1. In the Administration Console, click **Devices** > **Access Gateways** > **[Name of Server]** > **Alerts**.
2. To delete an alert from the list, select the check box for the alert, then click **Acknowledge Alert(s)**. To remove all alerts from the list, click the **Severity** check box, then click **Acknowledge Alert(s)**.

3 Click **Close**.

4 (Optional) To verify that the problem has been solved, click **Access Gateways** > **[Server Name]** > **Health** > **Update from Server**.

Viewing Access Gateway Cluster Alerts

1 In the Administration Console, click **Devices** > **Access Gateways** > **[Name of Cluster]** > **Alerts**.

2 Analyze the data displayed in the table.

Column	Description
Server Name	Lists the name of the Access Gateway that sent the alert. To view additional information about the alerts for a specific Access Gateway, click the name of an Access Gateway.
Severe	Lists the number of critical alerts that have been sent and not acknowledged.
Warning	Lists the number of warning alerts that have been sent and not acknowledged.
Information	Lists the number of informational alerts that have been sent and not acknowledged.

3 To acknowledge all alerts for an Access Gateway, select the check box for the Access Gateway, then click **Acknowledge Alert(s)**. When you acknowledge an alert, you clear the alert from the list.

4 To view information about a particular alert, click the server name.

Managing Access Gateway Alert Profiles

For an Access Gateway, this option allows you to send notification of generated system alerts to the Administration Console, to a Syslog server, to a log file, or to a list of e-mail recipients.

1 In the Administration Console, click **Devices** > **Access Gateways** > **Edit** > **Alerts**.

2 Select one of the following actions:

New: To add a new profile, click **New**. Specify a name for the profile, then click **OK**. For configuration information, see "Configuring an Alert Profile" on page 912.

Enable: To enable a profile, select the check box next to the profile, then click **Enable**.

Disable: To disable a profile, select the check box next to the profile, then click **Disable**.

Delete: To delete a profile, select the check box next to the profile, then click **Delete**.

3 Click **OK** twice.

4 On the **Access Gateways** page, click **Update**.

Configuring an Alert Profile

The alert profile determines which alerts are sent and where the alerts are sent.

1 In the Administration Console, click **Devices** > **Access Gateways** > **Edit** > **Alerts** > **[Name of Profile]**.

2 Select one or more of the following:

Connection Refused: Generated when a connection is refused.

Proxy Initialization Failure: Generated when the Embedded Service Provider fails to initialize.

System Up: Generated each time the Access Gateway is started.

System Down: Generated each time the Access Gateway is stopped.

Configuration Changed: Generated each time the configuration of the Access Gateway is modified.

Failure in Audit, Stopping Services: Generated when the audit server has failed, and the Access Gateway has been configured to stop services.

To configure the Access Gateway to continue when auditing services are not available, click **Auditing** in dashboard, deselect the **Stop Services on Audit Server Failure** option, then click **Apply**.

Failure in Audit, Will lose events, but continuing services: Generated when the audit agent has failed. The Access Gateway continues to run, but no audit events are generated.

As a workaround while solving this problem, you can enable proxy service logging (see "Configuring Logging for a Proxy Service" on page 861). The common and extended log files provide some details on the HTTP traffic.

If you do not want the Access Gateway to run without generating events, you need to manually shut down the Access Gateway.

Failure in Audit, Server is offline: Generated when the audit agent is unable to contact the audit server. When this condition occurs, the audit agent uses local caching for the audit events.

Do not allow this condition to continue indefinitely. The Access Gateway soon reaches the limits of its local cache. If this happens, events can be lost and the Access Gateway might need to stop services.

For troubleshooting information, see "Troubleshooting Novell Audit" (http://www.novell.com/documentation/novellaudit20/novellaudit20/data/al0lh30.html) in the *Novell Audit Administration Guide* (http://www.novell.com/documentation/novellaudit20/novellaudit20/data/bookinfo.html).

3 Select where you want the alerts sent:

Send to Device Manager: Select this option to send alerts to the Administration Console.

Send to SNMP: (Access Gateway Service) Select this option to send alerts to an SNMP server. To configure the SNMP server, click the **Send to SNMP** link. For configuration information, see "SNMP Profile" on page 914.

Send to Log File: Select this option to send alerts to a log file. To send alerts to a log file, click **New**, specify a name for the log profile, then click **OK**. For configuration information, see "Configuring a Log Profile" on page 914.

To enable a log profile, select the profile, then click **Enable**.

To disable a log profile, select the profile, then click **Disable**.

To delete a log profile, select the profile, then click **Delete**. Click **OK** in the confirmation dialog box.

Send E-mail Notifications: Select this option to send alerts through e-mail notifications. To enable e-mail notification click **New**, specify a name for the e-mail profile, then click **OK**. For configuration information, see "Configuring an E-Mail Profile" on page 914.

To enable an e-mail profile, select the profile, then click **Enable**.

To disable an e-mail profile, select the profile, then click **Disable**.

To delete an e-mail profile, select the profile, then click **Delete**. Click **OK** in the confirmation dialog box.

Send to Syslog: Select this option to enable Syslog alerts. Click **New**, specify a name for the Syslog profile, then click **OK**. For configuration information, see “Configuring a Syslog Profile” on page 915.

To enable a syslog profile, select the profile, then click **Enable**.

To disable a syslog profile, select the profile, then click **Disable**.

To delete a syslog profile, select the profile, then click **Delete**. Click **OK** in the confirmation dialog box.

4 To enable an alert action profile, select the action profile, click **Enable**, then click **OK**.

The action to send the alerts to a log file, to e-mail addresses, or to a syslog file is not performed until the action profile is enabled.

5 On the Alert Profiles page, verify that the alert profile you have created is enabled, then click **OK** twice.

6 Update the Access Gateway.

SNMP Profile

1 (Access Gateway Service) To add the IP address of a SNMP server, click **New**, specify the IP addresses, then click **OK**.

2 (Optional) To delete an IP address, select the IP address, then click **Delete** > **OK**.

3 Select one of the following:

- Add another profile, continue with Step 3 on page 913.
- Save your modifications, continue with Step 4 on page 914.

Configuring a Log Profile

The **Send to Log File** field displays the name of the log profile you are configuring.

1 Fill in the following fields:

Log File Name: Specify a name for the log file and a path where the file should be stored.

You must specify the full path.

```
/var/opt/novell/amlogging/logs/
```

Max File Size: Specify a maximum size for the log file in KB. The size can be from 50 to 100000 KB. Specify 0 to indicate that there is no maximum file size.

2 Click **OK**.

3 Select one of the following:

- To add another profile, continue with Step 3 on page 913.
- To save your modifications, continue with Step 4 on page 914.

Configuring an E-Mail Profile

The **Send E-Mail Notifications** field displays the name of the e-mail profile you are configuring.

1 Fill in the following fields:

E-mail Recipients: To add a recipient to the list, click **New**, specify the e-mail address of the recipient, then click **OK**. You can add multiple e-mail addresses. To delete a recipient, select the user’s email address, click **Delete**, then click **OK**.

Mail Exchange Servers: To add a mail server, click **New**, specify the IP address or the DNS name of the mail exchange server, then click **OK**. You can add multiple mail exchange servers. To delete a server, select the server, click **Delete**, then click **OK**.

2 Click **OK**.

3 Select one of the following:

- To add another profile, continue with Step 3 on page 913.
- To save your modifications, continue with Step 4 on page 914.

Configuring a Syslog Profile

The **Send to Syslog** field displays the name of the Syslog profile you are configuring.

1 Fill in the following field:

Facility Name: Specify a facility name for the Syslog server. It can be any name from local0 to local7. If you specify local0 - local7 as your facility name, the alerts are stored at `/var/log/localmessages`.

2 Click **OK**.

3 Select one of the following:

- To add another profile, continue with Step 3 on page 913.
- To save your modifications, continue with Step 4 on page 914.

To configure the Syslog profile for Access Gateway Service on RedHat Enterprise Linux, use the following procedure:

1 Go to `/etc/rsyslog.conf` file.

2 Add the following under `# Provides UDP syslog reception`

$ModLoad imudp.so

$UDPServerRun 514

3 Restart the syslog service using one of the following commands:

`/etc/init.d/rsyslog restart` OR `rcsyslog start`

23 Monitoring Access Manager By Using Simple Network Management Protocol

The Administration Console captures all statistics sent by the Identity Server and the Access Gateway. These statistics sent at periodic intervals, are stored in eDirectory.

You can use any Network Monitoring System (NMS) or an SNMP-enabled client to gather statistics from the Administration Console by using Simple Network Management Protocol (SNMP). Simple Network Management Protocol (SNMP) is a network management protocol for network management that collects information from devices on a network. Access Manager supports SNMP v2 for the purpose of monitoring Identity Server and Access Gateway.

NOTE: This release of Access Manager does not support SNMP traps.

SNMP Architecture in Access Manager

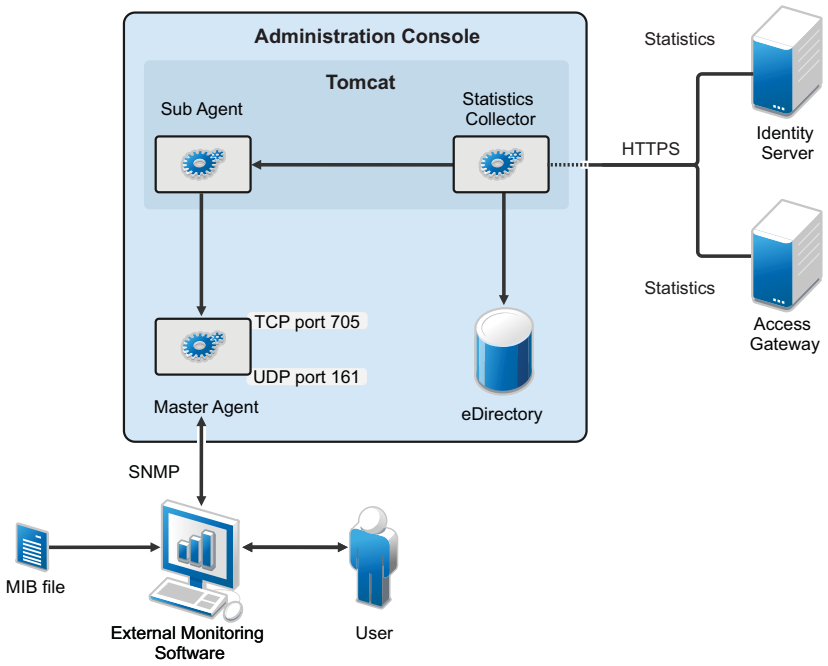
Access Manager introduces Master Agent, Sub Agent, and a Management Information base (MIB) file to work with any third-party monitoring software using SNMP.

The Master Agent runs as a service in the Administration Console and listens to the Sub Agents registered with it. A Sub Agent is a managed device that is registered with the Master Agent and exchanges information with it using TCP port 705. The MIB file contains a hierarchical list of variables and defines the information that is provided by the devices. Each variable in this list is uniquely identified by an OID (Object Identifier) and are read-only in nature.

The Administration Console contains both Master Agent and Sub Agent. Master Agent runs as a separate service and the Sub Agents are registered with the Master Agent for monitoring. The Administration Console gathers statistics from all devices and acts as a centralized repository for any monitoring tool to access the data by using SNMP. The external NMS contacts the Administration Console to get the data about any Identity Server or Access Gateway by using SNMP. For this communication it uses UDP port 161 (by default).

In a clustered Administration Console setup, the devices send statistics to the secondary Administration Console in case the primary Administration Console is down.

Figure 23-1 *Architecture of SNMP Components in Access Manager*



This MIB file contains all the Identity Server and Access Gateway attributes available to monitor the state of the system. Figure 23-1 on page 918 illustrates how Administration Console uses SNMP to monitor the Identity Server and the Access Gateway.

If you are installing or upgrading Access Manager on a Linux server, the Master Agent is automatically installed. A Windows server has an in-built SNMP Master Agent, but it does not support the AgentX protocol. The AgentX protocol is used for communication between the Master Agent and Sub Agent. Due to this, if you are installing Access Manager on a Windows server, the Master Agent has to be downloaded and installed manually. For more information about installing the Master Agent on a Windows server, see “Installing and Enabling Monitoring for Access Manager on Windows” on page 922

Features of Monitoring in Access Manager

In Access Manager 4.0, monitoring using SNMP includes the following features:

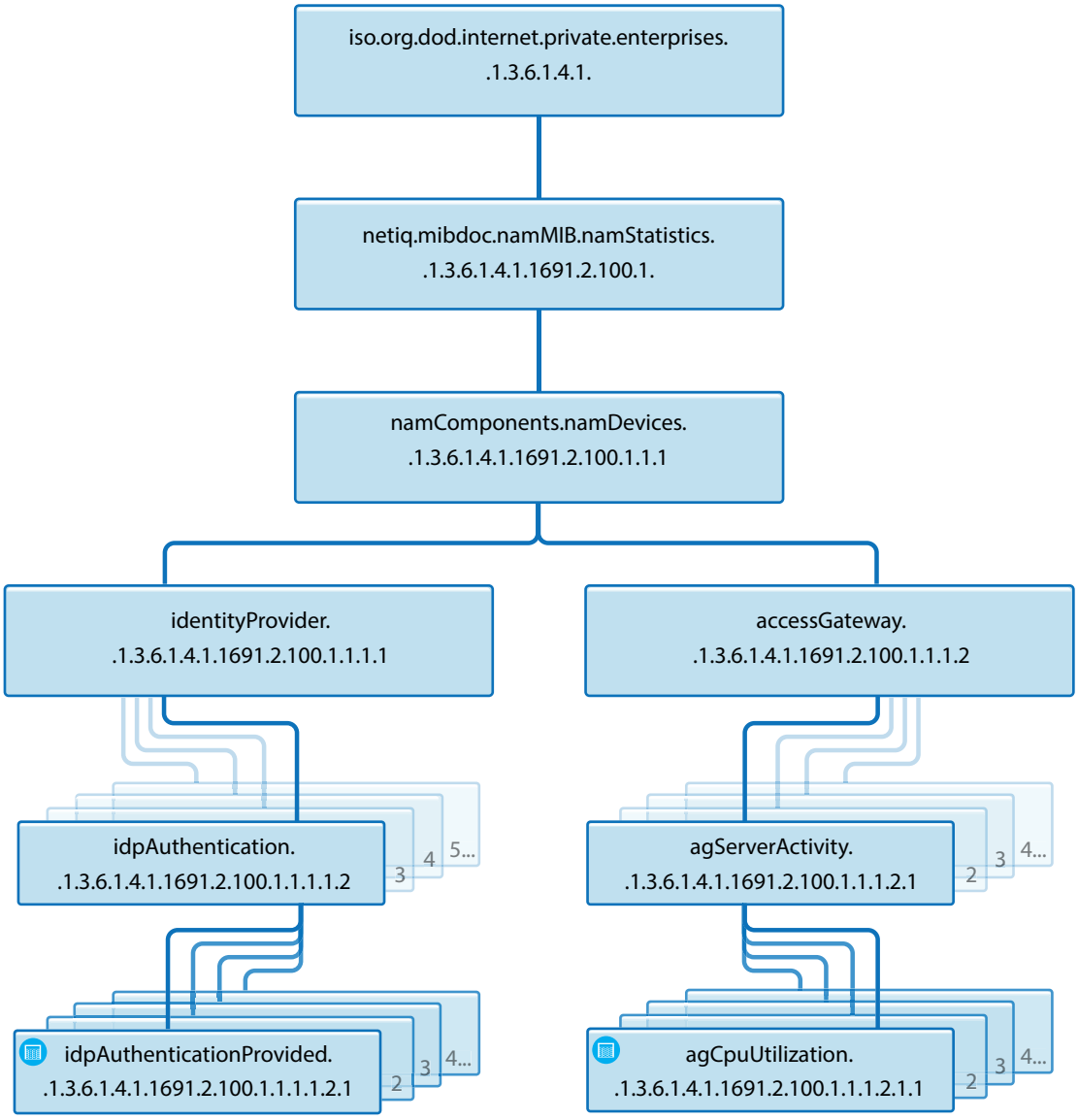
- Ability to enable/disable monitoring - By default SNMP is not enabled. You can configure it to enable monitoring for Access Manager components. For details about enabling monitoring, see “Installing and Enabling Monitoring for Access Manager Components” on page 921
- Facility to change port information or IP address of the Master Agent- You can configure the Master Agent to listen on a different port or IP address. The default port is TCP 705.
- Master Agent and Sub Agent architecture support multiple sub agents - The Master Agent - Sub Agent architecture helps you configure additional Sub Agents to be monitored. For example, you can configure a single Master Agent to receive data from Access Manager components, eDirectory as well as SLES Sub-Agents.
- Automatic data synchronization on device addition or removal - The MIB structure is automatically adjusted for dynamic addition or removal of components
- Automatic reconnect to Master Agent - Every time the Administration Console is restarted the reconnection to the MasterAgent happens automatically. No manual steps are required.

Using the Default MIB File with External SNMP Systems

When Access Manager is installed, `NAM.mib` file is placed in the `opt/novell/devman/share/conf` folder. On a Windows server this file is placed in the `C:\Program Files (x86)\Novell\Tomcat\webapps\roma\WEB-INF\` folder.

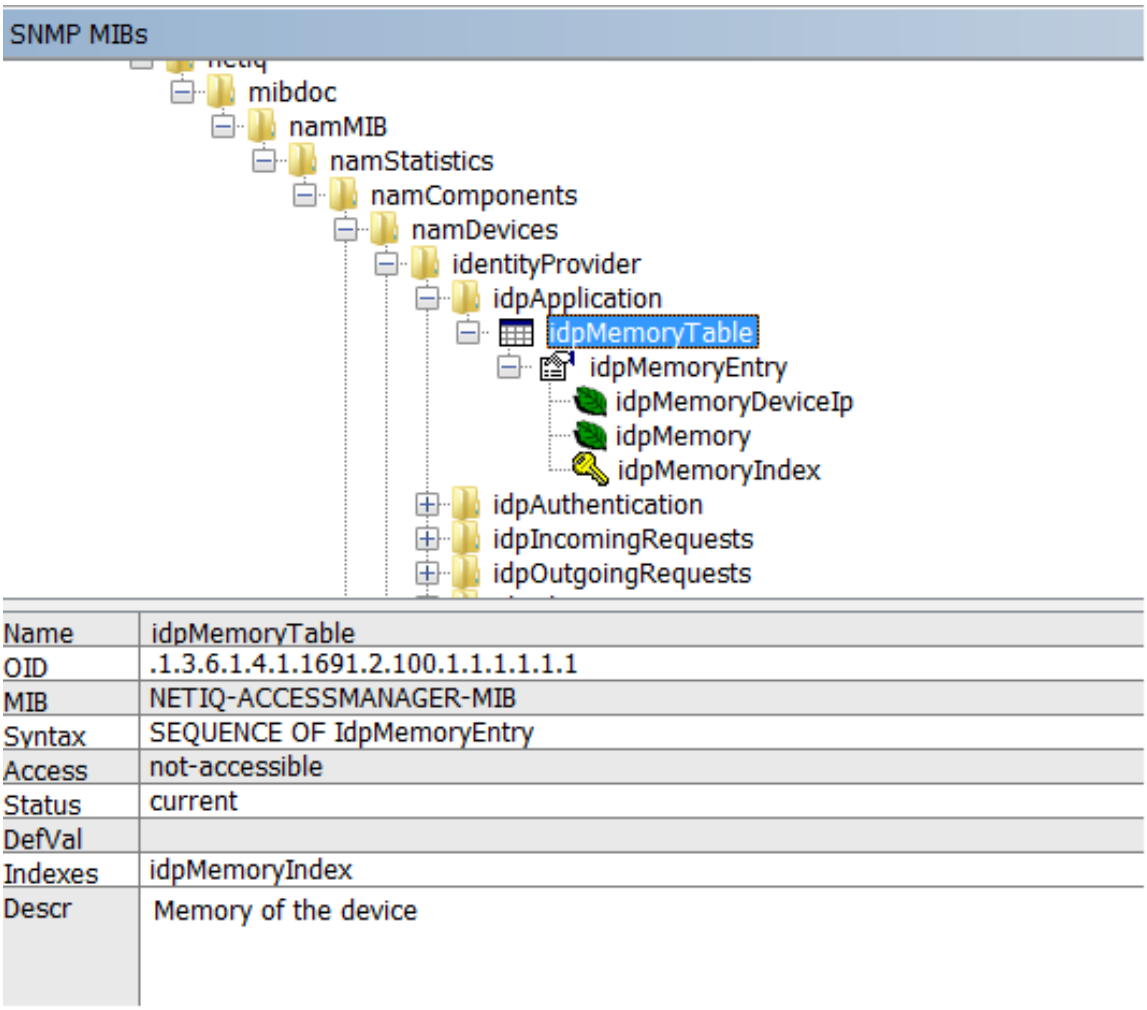
This MIB file contains textual information of Identity Server and Access Gateway attributes. You can use these attributes to monitor the state of the system. The attributes are uniquely identified by an OID (Object Identifier) or namespace. Figure 23-2 on page 919 shows the hierarchy of a MIB file when viewed with a MIB browser.

Figure 23-2 *The MIB file viewed in a MIB Browser*



Each statistic entry in the MIB file has a corresponding description to help identify the attribute.

Figure 23-3 *Description of an attribute in the MIB file*



Every time a new Identity Server or Access Gateway is added or removed the SNMP data available in the Administration Console is updated. If a device is not accessible for some reason, the MIB file (when viewed with a MIB Browser) displays the last reported statistics for all the attributes except for the Health Status of the devices. The Health Status of the devices are updated periodically.

Querying For SNMP Attributes

To query any SNMP attribute the following details are needed:

- IP Address of the Administration Console
- The community string name
- The Object Identifier (OID) of the attributeID
- The IP address of the device - Identity Server or Access Gateway.

For example, consider a scenario where you want to query the memory utilization of an Identity Server with IP address 10.0.0.0. The query is issued to the Administration Console whose IP address is 192.168.0.0

You can perform the query either by using the OID or by using the namespace of the object.

If you are using the net-snmp package for monitoring, the equivalent command to retrieve memory utilization details are:

Querying Using the Namespace

```
snmpget -v2c -m /opt/volera/roma/conf/NAM.mib -c netiq 192.168.0.0
.iso.org.dod.internet.private.enterprises.netiq
.mibdoc.namMIB.namStatistics.namComponents.namDevices.identityProvider.idpApplicat
ion.idpMemoryTable.idpMemoryEntry.idpMemory.10.0.0.0
```

NOTE: You must provide the exact path of the Access Manager mib file.

Querying Using the OID

```
snmpget -v2c -c netiq 192.168.0.0 .1.3.6.1.4.1.1691.2.100.1.1.1.1.1.1.1.10.0.0.0
```

In the same manner, you can query values of various attributes supported by the Identity Server and the Access Gateway.

Using the same example, if you query idpHealthEntry parameter by using the Namespace, the command is:

```
snmpget -v2c -m /opt/volera/roma/conf/NAM.mib -c netiq 192.168.0.0
.iso.org.dod.internet.private.enterprises.netiq.mibdoc.namMIB.namStatistics.namCom
ponents.namDevices.identityProvider.idpApplication.idpMemoryTable.idpMemoryEntry.i
dpMemory.10.0.0.0
```

The idpApplication parameter is substituted with the idpHealthEntry attribute in the above example.

NOTE: You must provide the exact path of the Access Manager mib file.

Understanding Return Values of an SNMP Query

When an SNMP query is performed, it retrieves the last fetched data from the Administration Console. If the device is down or not reachable a negative value is retrieved.

For example: If you query for the idpHealthyEntry attribute, the value that is returned can be Red, Yellow, Green or NoReport.

NOTE: The return value of NoReport indicates a server that is disconnected or unavailable.

Installing and Enabling Monitoring for Access Manager Components

- “Installing and Enabling Monitoring for Access Manager on Linux” on page 922
- “Installing and Enabling Monitoring for Access Manager on Windows” on page 922

Installing and Enabling Monitoring for Access Manager on Linux

1. To install the Master Agent and Sub Agent on Linux, no manual steps are required.

   All packages necessary to monitor Access Manager are automatically installed during upgrade or installation. The Administration Console is automatically installed and configured as the Master Agent and the Sub Agents are in turn registered with the Administration Console for monitoring.

2. In the `opt/novell/devman/share/conf/platform.conf` file, traverse to the `vcdn` module for SNMP. In `<stringParam name="enable" value="false"`, replace false with true and save the file using `:wq!`.

   This enables monitoring between Access Manager devices.

   The vcdn module also contains port details. If needed, you can configure the Master Agent to listen on a different port or IP address. The default port is TCP 705.

3. In the `snmp-master-agent.conf` file, change the community name. The default name is `netiq`. Changing the community name is recommended for security purpose.

4. Start the Master Agent by using the `/etc/init.d/novell-snmpd start` command.

5. Restart the Administration Console using `/etc/init.d/novell-ac restart` command for the changes to take effect.

6. If you encounter any errors while enabling monitoring, review the `platform.0.log` file available in the `/var/opt/novell/nam/logs/adminconsole/volera` folder.

Installing and Enabling Monitoring for Access Manager on Windows

1. On a Windows server, the Master Agent has to be manually installed and configured.

   Download the latest `net-snmp` package and install it. For downloading binaries, go to Sourceforge (The supported version is 5.6.1.1).

2. Register windows service by running the following command:

   ```
   C:\usr\bin\snmpd.exe -register -Lf "C:/usr/log/snmpd.log" -c "C:/Program Files (x86)/Novell/Tomcat/webapps/roma/WEB-INF/conf/snmp-master-agent.conf"
   ```

   If you uninstall `net-snmp`, it is important to unregister. Use the following command to unregister:

   ```
   C:\usr\bin\snmpd.exe -unregister -Lf "C:/usr/log/snmpd.log" -c "C:/Program Files(x86)/Novell/Tomcat/webapps/roma/WEB-INF/conf/snmp-master-agent.conf"
   ```

3. In the `C:\Program Files (x86\Novell\Tomcat\webapps\roma\WEB-INF\conf\platform.conf` file, traverse to the `vcdn` module.

   In `<stringParam name="enable" value="false"`, replace false with true. This enables monitoring between Access Manager devices.

   The `vcdn` module also contains port details. If needed, you can configure the Master Agent to listen on a different port or IP address. The default port is TCP 705.

4. In the `snmp-master-agent.conf` file, change the community name. The default name is `netiq`. Changing the community name is recommended for security purpose.

5. Start the Master Agent by using the net start `"Net-SNMP Agent"` command.

   NOTE: Ensure that you specify the command within quotes to start the Master Agent.

6 Restart the Administration Console for the changes to take effect.

7 If you encounter any errors while enabling monitoring, review the logs available in the `C:\Program Files (x86)\Novell\log\platform.0.log` folder.

If you are on a Windows 2008 R2 server (upgraded to Access Manager 4.0 from an Access Manager 3.x version), then enabling SNMP monitoring does not update the `platform.0.log` file.

To enable SNMP Monitoring and ensure `platform.0.log` file is updated, perform the following steps:

7a Stop Tomcat server.

7b Edit the `C:\Program Files (x86)\Novell\Tomcat\webapps\roma\WEB-INF\conf\platform.conf` file.

7c Traverse to the end of the `platform.conf` file and locate the last `</vcdnModule>` tag.

7d Add the following content to appear after the last `</vcdnModule>` tag

```
<vcdnModule name="snmp"
className="com.volera.vcdn.platform.snmp.SnmpAgentInit"
 sequence="3">
                <stringParam name="enable" value="true"/>
                <stringParam name="masterAgentIp" value="127.0.0.1"/>
                <stringParam name="masterAgentPort" value="705"/>
</vcdnModule>.
```

Ensure that this content is placed inside the `<vcdnApplicationModule>` tag.

7e Start the Tomcat server.

This ensures that SNMP Monitoring is enabled on a Windows 2008 R2 server and the `platform.log` file is also updated.

24 Back Up and Restore

You can run backup and restore utilities from the command line to back up and restore your Access Manager Appliance configuration. An additional script, Diagnostic Configuration Export, allows you to export your configuration so NetIQ Support can help diagnose possible configuration problems.

For more information about the Diagnostic Configuration Export utility, see “Diagnostic Configuration Export Utility” on page 950.

The following sections describe how to back up and restore your Access Manager Appliance configuration, how to export your configuration for NetIQ Support, and how to restore the configuration of Identity Servers and Access Gateways:

- “How The Backup and Restore Process Works” on page 925
- “Backing Up the Access Manager Appliance Configuration” on page 926
- “Restoring the Access Manager Appliance Configuration” on page 927

How The Backup and Restore Process Works

- “Default Parameters” on page 925
- “The Process” on page 925

Default Parameters

All scripts call the `getparams.sh` script to request the parameters from the user. The `defbkparm.sh` script is created by the Access Manager installation. It contains the default parameters for different options required by the underlying backup and restore utilities. If the entries in this file are commented out, the user is prompted for additional parameters.

The Process

The backup script must be run on the primary Administration Console. It creates a ZIP file that contains all certificates that various devices use and an encrypted LDIF file that contains configuration parameters for all imported devices. You do not need to back up the configuration of individual devices. By backing up the primary Administration Console, you back up the configuration of all Access Manager devices.

The backup script backs up objects in the ou=accessManagerContainer.o=novell container. It does not back up the following:

- Admin user account and password
- Delegated administrator accounts, their passwords, or rights
- Policy View user accounts, their passwords, or rights
- Role Based Services (RBS) configuration
- Modified configuration files on the devices such as the `web.xml` file

- Local files installed on devices such as log files
- Custom login pages, custom error pages, or custom messages

You need to perform you own backup of custom or modified configuration files.

For information on how to perform a configuration backup, see “Backing Up the Access Manager Appliance Configuration” on page 926.

You need to restore a backup when the Administration Console fails. If another device fails, you simply replace the hardware, reinstall the appliance using the IP address of the failed appliance, and the device imports into the Administration Console and acquires the configuration of the failed appliance.

If the Administration Console fails, you need to restore the configurations you backed up. Replace the hardware and reinstall the Administration Console by using the DNS name and IP address of the failed console. Then use the restore utility to restore the certificates and the device configuration. The Administration Console notifies all devices that it is online and they resume communicating with it rather than using a secondary console.

Backing Up the Access Manager Appliance Configuration

1. On the primary Administration Console, change to the utility directory.

   `/opt/novell/devman/bin`
2. Run the following command:

   `./ambkup.sh`
3. Specify and confirm the Access Manager administration password.
4. Specify a path to save the backup files.
5. Specify a password for encrypting and decrypting private keys, then re-specify it for verification.

   You must use the same password for both backup and restore.
6. Press Enter.

NOTE: After running the backup script, check the logs to verify that no errors occurred while running the backup script. The log file location is displayed at the end of the script execution.

The backup script creates a ZIP file containing several files including the certificate information. This file contains the following:

- The configurations store’s CA key.
- The certificates contained in the configuration store.
- The trusted roots in the trustedRoots container of the accessManagerContainer object.
- An encrypted LDIF file, containing everything found in the OU=accessManagerContainer,O=novell container.
- A `server.xml` file containing the Tomcat configuration information for the Administration Console.
- A “delegatedusers_list” file containing the details of delegated users.

- A “policyviewusers_list” file containing the details of delegated users.
- A “backup_info” file that contains the basic details of the system on which the backup is being taken.

The trusted roots are backed up in both LDIF and ZIP files. They are added to the ZIP file so that the ZIP file has the complete certificate-related configuration.

IMPORTANT: The backup utility prompts you for a location to store the backup file. Select a location from where the backup file will not be deleted when you uninstall the product. The default location is `/root/nambkup`.

Name of the backup zip file stores some information. Do not change the name.

NOTE: Whenever the configuration store contains a Key Material Object (KMO) with a certificate signing request in pending state, the KMO will not be exported by using the `amdiagcfg` script and not be backed up by using the `ambkup` script.

NOTE: For security purposes, delegated users, policy view users, and users in the trusted and configuration stores are not backed up. You need to recreate them while restoring the configuration. You can find the common name and full name of these users during the restore process or in the files in the zip file.

Restoring the Access Manager Appliance Configuration

The restore script replaces the existing configurations in the configuration database with the configuration in the backup of the configuration store. It should be used to restore configuration data in one of the following scenarios:

- An upgrade failed and you need to return to the configuration before the upgrade.
- You want to return to the backed up configuration because the current modified configuration does not meet your needs.

If the primary Administration Console machine has failed, you have lost both the configuration and the configuration database. To recover from this scenario, you need to do more than restore the configuration.

The restore script cannot be used to move the Administration Console to a different platform, even if the new machine is configured to use the same IP address and DNS name. The backup files contains path information that is specific to the operating system.

- “Restoring the Configuration on the Same Appliance for Which Backup Was Taken” on page 928
- “Restoring the Configuration on a Freshly Installed Appliance with Same IP Address and DNS Settings” on page 928

NOTE: Restore should be made on the same version that was used to take the backup.

Restoring the Configuration on the Same Appliance for Which Backup Was Taken

1. Ensure that the zip file created during the backup process is accessible.
2. Log in to as `root`.
3. Change the current directory to the utility directory: `/opt/novell/devman/bin`
4. Run the following command:

   `./amrestore.sh`
5. Specify and confirm the Access Manager administration password.
6. Specify the path where the backup file is available.
7. Specify the name of the backup file. Do not include the `.zip` extension.
8. Specify the private key encryption password, then press Enter.

   Confirm the private key encryption password, then press Enter.
9. Wait for the restore process to complete.
10. (Conditional) If you have a secondary appliance installed, reboot the machines.
11. (Conditional) If any devices report certificate errors, you need to re-push the certificates.
    - **11a** Click **Troubleshooting** > **Certificates**.
    - **11b** Select the store that is reporting errors, then click **Re-push Certificates**.

      You can select multiple stores at the same time.
    - **11c** (Optional) To verify that the re-push of the certificates was successful, click **Security** > **Command Status**.

Restoring the Configuration on a Freshly Installed Appliance with Same IP Address and DNS Settings

In this scenario, apart from restoring the Administration Console configuration, you need to re-import the device settings too.

1. Ensure that the zip file created during the backup process is accessible.
2. Log in to as `root`.
3. Change the current directory to the `/opt/novell/devman/bin` directory.
4. Run the following command:

   `./amrestore.sh`
5. Specify and confirm the Access Manager administration password.
6. Specify the path where the backup file is available.
7. Specify the name of the backup file. Do not include the `.zip` extension.
8. Specify the private key encryption password, then press Enter.

   Confirm the private key encryption password, then press Enter.

   Wait for the restore process to complete.
9. Change the current directory to the utility directory:

   `/opt/novell/devman/jcc`
10. Run the following command:

```
conf/reimport_nidp.sh jcc
```

11 Follow the steps to re-import the jcc settings.

Wait for jcc to start.

12 Run the following command:

```
conf/reimport_nidp.sh nidp
```

13 Follow the steps to re-import the Identity Server settings.

Wait for the Identity Server health to turn green. You can check this in the Administration Console Dashboard.

14 Run the following command:

```
conf/reimport_ags.sh agm
```

15 Follow the steps to re-import the Access Gateway settings.

Wait for the Access Gateway health to turn green. You can check this in the Administration Console Dashboard.

16 (Conditional) If you have a secondary appliance installed, reboot the machines.

17 (Conditional) If any devices report certificate errors, you need to re-push the certificates.

17a Click **Troubleshooting** > **Certificates**.

17b Select the store that is reporting errors, then click **Re-push Certificates**.

You can select multiple stores at the same time.

17c (Optional) To verify that the re-push of the certificates was successful, click **Security** > **Command Status**.

25 Code Promotion

Code Promotion helps you replicate the configuration data of Access Manager from one setup to another. You can export the configuration data as a password-protected encrypted file, then import this file into another Access Manager system and seamlessly replicate the configuration into the target system.

The exported configuration data includes generic Identity Server cluster configuration, customization files, proxy services, protected resources, and policy configuration. The exported data is independent of the device specific data and network specific data. Therefore, you can use Code Promotion to replicate configuration between two Access Manager systems that are in different networks, with a different number of devices, and with different user stores.

- "How Code Promotion Helps" on page 931
- "Sequence of Promoting the Configuration Data" on page 932
- "Prerequisites" on page 932
- "Limitations" on page 933
- "Configuring Custom File Paths" on page 933
- "Exporting the Configuration Data" on page 934
- "Importing the Configuration Data" on page 935
- "Troubleshooting Code Promotion" on page 942

How Code Promotion Helps

Code promotion helps you seamlessly perform the following activities:

- **Managing multiple Access Manager setups:** When managing multiple Access Manager setups, you may have to replicate the same configuration in all setups.

  For example, you want to test your configuration in a dedicated staging setup and then build a new production setup based on the tested configurations. Or, you maintain multiple staging setups and you want the configuration changes to pass through on these setups before deploying the configuration data to an existing production setup. You do not need to manually replicate the configuration data in other setups.

  Code Promotion provides a mechanism to move the configuration data across Access Manager setups. Code promotion increases efficiency, improves productivity, and in turn reduces costs of managing configurations across environments.
- **Managing different setups by different administrators:** Different administrators can manage different Access Manager environments. Manually replicating the configuration to different setups requires maintenance of a precise list of all changes done on one system and this knowledge must be transferred among administrators. Code Promotion takes all configuration changes and replicates correctly.
- **Replacing or moving physical devices:** You may need to replace physical devices or move them to a different network due to a business decision, such as changing a network infrastructure vendor. For example, you want to move your application to another physical server

or you want to move the application hardware to a different network infrastructure. Code Promotion is independent of network-specific changes and helps you easily transfer the configuration data.

- **Adding devices in the cluster:** You have added a device in a cluster for capacity needs. When you add a device to a cluster, Access Manager applies the customization of that cluster to the device.
- **Adding a new application or path:** You have added a new application or a new path and you want to replicate it to another environment.
- **Adding or modifying a protected resource:** You have added or modified a protected resource and you want to replicate it to another environment.

Sequence of Promoting the Configuration Data

You must move the configuration data in the following sequence:

1. Identity Server configurations, policies configurations, Certificates and Keystores configurations, and Identity Server custom files
2. Access Gateway configurations and Access Gateway custom files

---

NOTE: If you want to import only protected resources or proxy services along with the related Identity Server contracts, you need to import only the Access Gateway configuration. Code Promotion also imports the Identity Server dependencies.

Prerequisites

Ensure that you have read and implemented the following prerequisites before performing Code Promotion:

- The source server and the target server must have the same version of Access Manager. If you want to export the configuration data from an earlier version of Access Manager, ensure that you first upgrade Access Manager on the source server to the Access Manager version installed on the target server. For more information about how to upgrade Access Manager, see the "Upgrading Access Manager Appliance" in the *NetIQ Access Manager Appliance 4.2 Installation and Upgrade Guide*.
- The source server and the target server must run on the same operating system.
- The source server and the target server must have the same model; that is, both must be either Access Manager or Access Manager Appliance.
- Importing configuration data replaces the existing configuration data. Therefore, use the backup option in the Import wizard to preserve a copy of the existing configuration before importing the data.
- Before you import the Access Gateway configuration, you must manually create reverse proxies and master proxy services or root proxy service on the target system.
- Each configuration entity that you want to map between source and destination systems should have the same name. Configuration entities include proxy service, protected resource, authentication class, method, contract, and user stores.

- Back up the Access Gateway configuration by using the `ambackup` file. For more information, see Chapter 24, “Back Up and Restore,” on page 925.

  The backup option available on the Code Promotion user interface works only for the Identity Server configuration.

Limitations

The following list includes limitations in Code Promotion:

- Code Promotion supports export and import of only the generic configuration data. It does not support export and import of the configuration data that vary from one system to another. For example, you cannot export and import network specific configuration, device specific configuration, configuration store, and its replica ring configuration.
- Customization files that you want to import should contain only generic information, not any device-specific information. For example, `server.xml` contains local keystore passwords. Therefore, you should not apply it to all devices.
- For Access Gateway, Code Promotion supports export and import of only proxy services and protected resources along with configured contracts and policies.
- Code Promotion does not support export or import of only custom files.
- Internet Explorer Compatibility View does not support Code Promotion.
- Code Promotion takes a significantly longer time on Windows, especially importing the metadata repository. You must wait until the import process is complete to avoid data corruption.
- Code Promotion does not support export or import of Appmarks configuration data.

Configuring Custom File Paths

Access Manager provides a configurable list of files and directories for the Identity Server and the Access Gateway to export. This list contains default paths of the most frequently customized files. If you have customized any additional files or you have saved the custom file at any other location instead of the default directory, you must update the path before initiating the export process.

This list also contains paths of customized files for Access Manager installed Windows. You can ignore the paths if your setup is on Linux.

Perform the following steps to configure the custom file paths:

1. In the Administration Console Dashboard, click icon at the top right of the page and then click **Code Promotion** > **Settings**.

   Use this tab only when you export the configuration data. You do not need to make any change in file paths while importing the configuration data.
2. Ensure that the paths are correct. Update the default paths with the actual paths wherever applicable.

   NOTE: You must provide the complete name of the custom file. Code Promotion does not support wildcard characters in file names. For example:

   Supported: `/opt/novell/nam/mag/webapps/agm/WEB-INF/config/current/ErrorMessages.xml.en`

Not supported: `/opt/novell/nam/mag/webapps/agm/WEB-INF/config/current/ErrorMessages.xml.*`

3 Click **OK**.

NOTE: Ensure that the custom files do not include any system-specific data.

Exporting the Configuration Data

You can download previously exported configuration files. Access Manager saves these exported files on the primary Administration Console system also at the following location:

`/var/opt/novell/novlwww/namconfig`

You can delete or back up these files if needed. If you delete these files from the disk, the Code Promotion page does not list them any longer.

The exported configuration data includes:

- Identity Server configuration
  - Cluster configuration
  - Shared Settings
  - Identity Server policies
  - Customization files
  - Risk-based authentication configuration
- Access Gateway configuration
  - Proxy services and protected resources
  - Access Gateway policies
  - Customization files

  NOTE: You cannot export configuration of an Access Gateway that is not part of any Access Gateway cluster.

Perform the following steps to export the configuration data:

1 Log in to the Administration Console from where you want to export the configuration data.

2 In the Administration Console Dashboard, click icon at the top right of the page and then click **Code Promotion**.

3 In the Code Promotion page, click **Export Configuration**.

4 Based on your requirements, select the configuration to export:

Identity Server Configuration: Exports all clusters, shared settings, keystores, trust stores, and Identity Server policies. You can also select to export Identity Server customization files, if any.

Access Gateway Configuration: Exports proxy services, protected resources, and Access Gateway policies. You can also select to export Access Gateway customization files, if any. Code Promotion exports all Identity Server dependent configurations, such as contracts assigned to protected resources, even though you selected only Access Gateway configuration to export.

If you want to export customization files, select respective devices to export customization files.

NOTE

- If you saved a customization file at a location that is not a default location, ensure that you update the file name, directory name, and path before exporting the file. For more information, see “Configuring Custom File Paths” on page 933.
- Code Promotion does not support import or export of only custom files.

5 Click **Next**.

6 (Optional) Specify a password to encrypt the archived configuration data file.

You require this password to decrypt the ZIP file while importing configuration data into another environment.

7 Click **OK** and save on your local system.

Importing the Configuration Data

You can import the configuration data either for Identity Server or for Access Gateway at one time. You need to repeat the process to import the configuration data of each component.

If you are importing the configuration data on a new production environment, you must import the Identity Server configuration, and create reverse proxies and master proxy services before importing the Access Gateway configuration data.

Import the configuration data only on the primary Administration Console. Importing the configuration data includes the following actions:

- “Uploading Configuration File to Import” on page 935
- “Selecting the Component to Import the Configuration Data” on page 936
- “Importing the Identity Server Configuration Data” on page 936
- “Importing the Access Gateway Configuration Data” on page 937
- “Post-Import Configuration Tasks” on page 940

Uploading Configuration File to Import

Perform the following steps to import the configuration data:

1 Log in to the Administration Console where you want to import the configuration data.

2 In the Administration Console Dashboard, click icon at the top right of the page and then click **Code Promotion**.

3 In the Code Promotion page, click **Import Configuration**.

4 Click **Browse** to import the configuration file.

5 In **Decryption Password**, specify the password that you used to encrypt the configuration data file. You need this password to extract the contents of the configuration file.

6 (Optional) Select **Backup current configuration before import** and **Backup customization files**. This backup helps to roll back your changes if needed. Code promotion encrypts the backup file with the same password that you specified for decryption in Step 5. You can download this backup file from the Code Promotion page.

NOTE: This option backs up only the Identity Server-specific configuration. To back up the Access Gateway configuration, you must use the `ambackup` file.

7 Click **Next**. Continue with "Selecting the Component to Import the Configuration Data" on page 936.

Selecting the Component to Import the Configuration Data

Code Promotion automatically detects whether the imported ZIP file contains configuration data of the Identity Server, Access Gateway, or both. It also checks for any device customization files.

1 Under **Select Configuration To Import**, select the option you need based on your requirements:

- **Identity Server Configuration:** Select this option to import the Identity Server configuration data. Select **Customization Files on Devices** if you want to import Identity Server customization files.
- **Access Gateway Configuration:** Select this option to import the Access Gateway configuration data. Select **Customization Files on Devices** if you want to import Access Gateway customization files.

2 (Only for Access Gateway) Under **Access Gateway Cluster Mapping**, specify the cluster in **Source Cluster** from which you want to export the configuration data and select the cluster in **Target Cluster** in which you want to import the configuration data. You can import configuration data of only one cluster at a time. If you want to import configuration from multiple clusters, run the import process separately for each cluster.

3 Click **Next**.

4 Continue with any one of the following sections based on the configuration you selected to import:

- "Importing the Identity Server Configuration Data" on page 936.
- "Importing the Access Gateway Configuration Data" on page 937.

Importing the Identity Server Configuration Data

Importing the Identity Server configuration data includes the following steps:

1. Uploading Configuration File to Import
2. Selecting the Component to Import the Configuration Data
3. Importing Identity Server Clusters
4. Post-Import Configuration Tasks

Importing Identity Server Clusters

1 In the **Import Identity Server Clusters** section, specify the import action for each cluster available in the imported configuration. Select the desired options based on your requirements.

NOTE: Importing Identity Server Configuration overwrites the existing Shared Settings on the system with new Shared Settings. However, if any of the existing settings on the target system are not part of the source system configuration, Code promotion will not delete them.

The following table lists examples with Attribute Sets and import action:

Imported Attribute Sets	Existing Attribute Sets	Import Action
OIOSAML with five mappings	OIOSAML with two mappings	Replaces OIOSAML set with the imported one. It has five mappings.
AttrSet1	Not available	Adds AttrSet1.
No import	AttrSet2 is defined only in the target system	AttrSet2 remains unchanged.

2 In **Clusters To Import**, select a cluster to configure import settings.

3 Select an action for the selected cluster from **Import Action**.

- **Import As New Cluster:** Select this option if you want to import the cluster as a new cluster. Ensure that the new cluster name is different from the existing cluster names defined on that system.
- **Overwrite Existing Cluster:** Select this option if you want to overwrite the existing cluster with the selected cluster.

NOTE: You need to configure the import action for each cluster separately. If the cluster you want to import has only one user store, Code Promotion maps the user store to the default user store of the existing cluster. If the cluster you are importing has multiple user stores, then you must specify how to map them to the user stores of the existing cluster.

4 Click **Next**.

Continue with "Post-Import Configuration Tasks" on page 940.

Importing the Access Gateway Configuration Data

Code Promotion uses names to associate entities from the source system to the target system. It searches on the source system for names that are part of the import. If it finds the Access Gateway entities with the same names, it overwrites these entities. If not available, it creates new entries with the same names from the source system. When the Identity Server and policies-specific entities with the same names are available, you can select whether to overwrite these.

If the policy name, policy extension, and proxy service match on the source and target systems, but their type does not match, then the import does not happen.

Code Promotion does not export Access Gateway clusters, reverse proxies, and master proxies. Before importing the Access Gateway configuration data, you must manually create clusters, reverse proxies, and master or root proxy services in the target system.

If you want to import the Access Gateway protected resources that require Identity Server configuration other than contracts and its dependencies, LDAP attributes, and Shared Secret, you must first import the required Identity Server configuration. For example, for risk-based authentication or OAuth configuration, you need to import relevant Identity Server configuration separately. You can import these configurations manually or by using Identity Server Code Promotion.

NOTE: If the reverse proxy in the source system is non-HTTP and in the target system it is HTTPS or vice-versa, ensure that you have tested the configuration before importing. In this case, the import may result in issues if there is any issue in the browser to Access Gateway communication.

Importing the Access Gateway configuration data includes the following steps:

1. Uploading Configuration File to Import
2. Selecting the Component to Import the Configuration Data
3. Selecting Proxy Services and Protected Resources to Import
4. Verifying the Component-Specific Configuration Changes
5. Updating Identity Server User Store References
6. Setting Up New Proxy Services in the Target System after Import
7. Post-Import Configuration Tasks

Selecting Proxy Services and Protected Resources to Import

When you select a proxy service for import, all protected resources associated with this proxy service are selected automatically. You cannot deselect any protected resources of a selected proxy service for import.

Code Promotion validates the content you want to import in to the target system. If there is any issue, it displays validation errors.

Code Promotion imports the Access Gateway customization details if you have selected the option. If any issue happens during customization files import, the system displays a message. You can continue or cancel the import process at that point.

To select proxy services and protected resources to import, complete the following steps:

1 The Code Promotion page displays the entire list of proxy services and protected resources from the source setup. Select proxy services and protected resources that you want to import.

2 Click **Next**. Continue with "Verifying the Component-Specific Configuration Changes" on page 938.

Verifying the Component-Specific Configuration Changes

Verify the details of configuration data that will be newly created and the data that will be overwritten on the destination system after import is complete. A proxy service may have a reference to logging profiles or http rewriter profiles. A protected resource refers to Identity Server contracts and policies. Identity Server contracts in turn refer to authentication class, methods, image sets, and user stores. A policy has a dependency on policy extensions, policy containers, Identity Server LDAP attributes and shared secrets. When you import the Access Gateway configuration, all of these dependencies are imported.

IMPORTANT: You can import only enabled rewriter and logging profiles, not the disabled profiles.

Regardless of the type of logging profile (common or extended) and rewriter profile (word or character), if the name of the profile is same on both the source and target systems, Code Promotion overwrites the profile.

To verify configuration changes, perform the following steps:

1 Select **Access Gateway** to verify the details about proxy services, protected resources, rewriter profiles, logging profiles, authentication procedures, and Access Gateway certificates that you are importing.

If you are importing a proxy service to a production setup where the same proxy service exists, the system will not overwrite the following parameters and will retain these:

- Published DNS Name
- Host Header
- Web Server Host Name
- Connect Port
- Web Server List

Access Manager locks the Access Gateway cluster and policy containers and releases these only after the import is complete or if you cancel the process before completing import.

2 Select **Identity Server** to verify the details about Identity Server contracts, methods, classes, LDAP attributes, shared secrets, and images that Code Promotion is importing along with the Access Gateway configuration data. Select **Overwrite Existing Contracts** if you have made any changes in the existing configuration in the source system. Selecting this option overwrites the contracts and their dependencies, such as methods and classes, in the target system. If you do not select to overwrite, Code Promotion does not import the modified configurations to the target system.

3 Select **Policy** to verify the details about policies, such as policy container and policy extension, that Code Promotion is importing along with the Access Gateway configuration data. Code Promotion matches policy containers by names for importing policies. If the names do not match, it creates new policy containers with that name on the target system. Select **Overwrite Existing Policies** if you have made any change to the existing configuration in the source system. Selecting this option overwrites the policies and its dependencies (such as policy extension, LDAP attribute, and shared secret) in the target system. If you do not select to overwrite, Code Promotion does not import the modified configurations to the target system.

   After selecting **Overwrite Existing Policies**, LDAP attributes and Shared Secret values in the Identity Server overview page may change. Verify the details and select **Verified** again on the Identity Server overview page.

4 Select **Verified** in each section.

5 Click **Next**. Continue with "Updating Identity Server User Store References" on page 939.

Updating Identity Server User Store References

If you have selected to overwrite a method or you have any new method that refers to a user store, update the reference of the user store of the source system to the user store of the target system. You can see the option to update user store references only when you select to overwrite a method or importing a new method.

You cannot reference the same user store on the target system to multiple user stores on the source system.

If the name of the user store on the source and target systems is the same, then the target system displays only that user store name that you should select.

If you have created a new user store in the source system, Code Promotion imports only the name to the target system. You must add entries manually after completing the import process.

To update the user store reference on the target system, perform the following steps:

1 Select the user store in **Imported User Store** and then select a corresponding user store in the target system under **Existing User Store**. Perform this activity for all imported user stores.

2 Click **Next**.

Continue with "Setting Up New Proxy Services in the Target System after Import" on page 940.

Setting Up New Proxy Services in the Target System after Import

To set up new proxy services in the target system, perform the following steps:

1 Specify the following details for all newly created proxy services:

NOTE: By default, all fields (Published DNS Name, Cookie Domain, Host Header, Web Server Host Name, Web Server List, and Connect Port) contain source system entries.

Published DNS Name: (Only for domain-based proxy services) Specify the DNS name you want the public to use to access your site. This DNS name must resolve to the IP address you set up as a listening address on the Access Gateway. The DNS name should be unique and not in use by any other proxy service.

Cookie Domain: Specify the domain for which the cookie is valid. Cookie domain is set as the corresponding master proxy service's cookie domain for domain-based and path-based proxy services. For a virtual proxy service, you can select a cookie domain based on the DNS specified.

Host Header: Specify the name you want to send in the HTTP header to the Web server.

Web Server Host Name: Specify the DNS name of the Web server that the Access Gateway should forward to the Web server.

Web Server List: Specify the Identity Server address or DNS name of Web servers. You can define it on cluster level. If you want to specify it for an individual server, go to **Devices** > **Access Gateways** > **Edit** > [*Name of Reverse Proxy*] > [*Name of Proxy Service*] > **Web Servers**. You can specify a **Web Server Host Name** for an individual server. For more information, see "Configuring Web Servers of a Proxy Service" on page 100.

Connect Port: Specify the port that the Access Gateway uses to communicate with the Web server.

2 Click **Next.**

3 Click **Finish** when the import process is completed. Continue with "Post-Import Configuration Tasks" on page 940.

Post-Import Configuration Tasks

After importing the Identity Server and Access Gateway configuration data, you must perform configurations that are specific to the target system and that are not part of the exported data.

Tasks after importing Identity Server configuration data

- After the import process is complete, the system displays a list of certificates that you need to create or import manually and apply. Code promotion imports Identity Server key stores, but you must create the certificates referenced in them on the server where you have imported the configuration data.
  - To create certificates, go to **Security** > **Certificates**. For more information about how to create certificates, see Section 11, "Creating Certificates," on page 799.

- The new certificate name must exactly match the names listed.
- Update Identity Server devices in the modified clusters. Go to **Troubleshooting** > **Certificates** and click **Re-push certificates**, and then update all devices in the cluster.

- Configure user stores for the newly added clusters. After the import process is complete, the system displays a list of Identity Server clusters for which you need to configure user stores. Code Promotion creates a placeholder entry for the user store. Code Promotions sets eDirectory as the default user store. You must enter the IP address, search context, and the password for the user stores of the target system. For more information, see "Configuring Identity User Stores" on page 272.
- For a newly added cluster, you need to manually add the Identity Server devices to it. This will enable you to use the imported configuration.
- Distribute the policy extension JARs to devices in the Administration Console under **Policy** > **Extensions**. For more information, see "Distributing a Policy Extension" on page 600.
- (Conditional) Update service providers with the new metadata. The identity provider certificate is different in the exported and imported systems. Therefore, you must re-import the identity provider metadata to all service providers in that cluster for federation to work. For more information, see "Viewing and Reimporting a Trusted Provider's Metadata" on page 158.
- Code Promotion does not import persistent federation identities and shared secrets. Only the Identity Servers in your exported setup and service providers share these. You must configure these on the server after you import the configuration data.
- When you add a new node in a cluster and no cache exists, the system takes customization of any active node in that cluster and applies that customization to this node on the target system. Modify the list of customization files to include all files as of the source setup. Otherwise, the customization available on the target system will be applied to the node.
- In case of User Attribute Retrieval and Transformation feature, after the import process is complete:
  - If a data source entry exists only in staging, then a new entry is created in the production environment. Code Promotion creates a placeholder entry for the data source fields. You must enter the username, password, IP, port, search context for LDAP, and URL of the data source
  - If a data source entry exists in the staging and the production environment, and, if the data source name is the same but has a different data source type, then, the production entry is retained.

Tasks after importing Access Gateway configuration data

- After the import process is complete, the system displays a list of certificates that you need to create or import manually and apply. Proxy key stores are imported, but you must create the certificates referenced in them on the target system.
  - To create certificates, go to **Security** > **Certificates**. For more information about how to create certificates, see Section 11, "Creating Certificates," on page 799. For more information about how to create certificates, see Section 11, "Creating Certificates," on page 799.
  - The new certificate name must exactly match with names listed.
  - Go to **Troubleshooting** > **Certificates** to re-push certificates and then update all devices in the cluster.
- If SSL is enabled between the imported proxy services and the web servers, and you selected to verify the certificate authorities of the web server certificates, then ensure that the web server's trusted roots are added to the Access Gateway's proxy trust store.

Go to **Troubleshooting** > **Certificates** to re-push certificates and then update all devices in the cluster.

- Configure the user store if you have imported a new user store. Configure or edit the user stores for the Identity Server clusters associated with the target Access Gateway cluster.
- Update the following Identity Server dependencies of policies with appropriate Identity Server cluster names and data if any of the policies refer to these:
  - Authentication contract, Liberty user profile, LDAP OU, Roles, LDAP group, credential profile, OAuth scope, and OAuth claims
  - Java data injection modules (these are deprecated)
- If you have imported the policy extensions, distribute the policy extension JARs to the devices in Administration Console under **Policy** > **Extensions** and restart the Access Gateway. If you imported policy extensions as part of Device Customization, then only restart the Access Gateway.

  For more information, see “Distributing a Policy Extension” on page 600.
- When you add a new node in a cluster and no cache exists, the system takes customization of any active node in that cluster and applies that customization to this node on the target system. Modify the list of customization files to include all files as of the source setup. Otherwise, the customization available on the target system will be applied to the node.
- If the imported Access Gateway components or policies refer to anything other than the following Identity Server dependencies, then, you must import these dependencies manually by using Identity Server Code Promotion: Contracts, methods, classes, user stores, LDAP attributes, and shared secrets.

Troubleshooting Code Promotion

See “Troubleshooting Code Promotion” on page 1016.

26 Troubleshooting

The following sections contain information about troubleshooting installation, upgrade, components of Access Manager, policies, and certificates:

- “Troubleshooting Installation” on page 943
- “Troubleshooting Upgrade” on page 945
- “Troubleshooting the Administration Console” on page 946
- “Troubleshooting the Access Gateway” on page 961
- “Troubleshooting Identity Server and Authentication” on page 981
- “Troubleshooting Certificate Issues” on page 1000
- “Troubleshooting Access Manager Policies” on page 1004
- “Troubleshooting MobileAccess” on page 1015
- “Troubleshooting Code Promotion” on page 1016
- “Troubleshooting OAuth and OpenID Connect” on page 1020
- “Troubleshooting User Attribute Retrieval and Transformation” on page 1023
- “Using Log Files for Troubleshooting” on page 1024
- “Access Manager Audit Events and Data” on page 1045
- “Event Codes” on page 1085

Troubleshooting Installation

- “Checking the Installation Logs” on page 943
- “Some of the New Hardware Drivers or Network Cards Are Not Detected during Installation” on page 944
- “Installation Through Terminal Mode is not Supported” on page 944
- “Novell Device Manager Installation Fails During the Appliance Installation” on page 945
- “Access Manager Appliance Installation Fails Due to an XML Parser Error” on page 945
- “DN Is Added as Provider ID While Installing NMAS SAML Method” on page 945

Checking the Installation Logs

If Access Manager Appliance fails to install, check the installation logs.

The installation logs are located in the `/tmp/novell_access_manager` directory. The following log files should contain useful content. Check them for warning and error messages.

Log File	Description
`install_main_2011-06-06_17:28:19.log`	Contains messages generated for installing and configuring Access Manager Appliance.

Log File	Description
`iinstall_edir_2011-06-06_17:38:35.log`	Contains messages generated for installing and configuring the Administration Console configuration store.
`install_audit_2011-06-06_17:38:35.log`	Contains messages generated for installing and configuring NetIQ Auditing components.
`Novell_iManager_2.7_InstallLog.log`	Contains messages generated for installing and configuring iManager.
`install_iman_2011-06-06_17:38:35.log`	Contains messages generated for installing and configuring iManager.
`install_adminconsole_2011-06-06_17:38:35.log`	Contains messages generated for installing and configuring the Administration Console.
`install_jcc_2011-06-06_17:38:36.log`	Contains messages generated for installing and configuring the Communications module.
`install_mag_2011-06-06_17:38:37.log`	Contains messages generated for installing and configuring the Access Gateway.
`install_idp_2011-06-06_17:38:36.log`	Contains messages generated for installing and configuring the Identity Server.
`configure_cluster_2011-06-06_17:28:19.log`	Contains messages generated for configuring Identity Server and Access Gateway.

Some of the New Hardware Drivers or Network Cards Are Not Detected during Installation

Installation of Access Manager Appliance might fail if some of the hardware drivers or network cards are not detected. If this happens, you must upgrade the hardware drivers manually:

1. Start the Access Manager Appliance installation.

   See Installing Access Manager Appliance in the NetIQ Access Manager Appliance 4.2 Installation and Upgrade Guide.
2. Select **Kernel Module (Hardware Driver)** in the main menu, then click **OK**.
3. Select **Add Driver Update**, then click **OK**.
4. Select the driver update medium.

   The driver update medium can be CD-ROM or floppy disk.
5. Click **OK**.

   The hardware driver is updated.
6. Continue with the Access Manager Appliance installation.

Installation Through Terminal Mode is not Supported

Installation through terminal mode is supported on GUI mode only. To work around this issue, initiate the installation in the GUI mode. After entering the required input, switch to the terminal mode. The installation is completed successfully.

Novell Device Manager Installation Fails During the Appliance Installation

To workaround this issue, reinstall the appliance.

Access Manager Appliance Installation Fails Due to an XML Parser Error

This error may happen if the Appliance is installed by using a remotely mounted installer. Use a locally mounted installer to avoid this issue.

DN Is Added as Provider ID While Installing NMAS SAML Method

While installing the NMAS SAML method in an external user store, DN is added as Provider ID instead of the metadata URL.

To resolve this issue, perform the following steps:

1. Log in to the Administration Console which has the external user store.
2. Go to **Roles and Tasks** > **NMAS** > **NMAS Login Methods** > **SAML Assertion** > **Affiliates**.
3. Select the respective Affiliate and change the provider ID to the identity provider metadata URL.

   For example, https://www.trunk2.com:8443/nidp/idff/metadata.

Troubleshooting Upgrade

- "The Access Gateway Throws a 403 Forbidden Page Error for a Resource Protected by a Form Fill Policy" on page 945
- "DN Is Added as Provider ID While Installing NMAS SAML Method" on page 946
- "Issue in SSL Communication between the Access Gateway and Web Applications" on page 946
- "Customized Login Pages Are Missing After Upgrading Access Manager" on page 946

The Access Gateway Throws a 403 Forbidden Page Error for a Resource Protected by a Form Fill Policy

This issue can happen if a Web server returns a form with a HTTP 403 error code. The Access Gateway, by default, returns its own custom error pages. Hence, this prevents the Form Fill feature to work. To workaround, go to **Access Gateway** > **Advanced Options**, enter ProxyErrorOverride off > click **OK**.

DN Is Added as Provider ID While Installing NMAS SAML Method

While installing the NMAS SAML method in an external user store, DN is added as Provider ID instead of the metadata URL.

To resolve this issue, perform the following steps:

1. Log in to the Administration Console which has the external user store.
2. Go to **Roles and Tasks** > **NMAS** > **NMAS Login Methods** > **SAML Assertion** > **Affiliates**.
3. Select the respective Affiliate and change the provider ID to the identity provider metadata URL. For example, https://www.trunk2.com:8443/nidp/idff/metadata.

Issue in SSL Communication between the Access Gateway and Web Applications

After upgrading Access Manager from 3.1 SP4 or 3.1 SP5 to 4.0.x, applications are not accessible. This issue happens when there is any discrepancy between the cipher suites configured for the Access Gateway and the applications protected by this Access Gateway. To workaround this issue, see TID 7016872.

Customized Login Pages Are Missing After Upgrading Access Manager

After upgrading Access Manager, you cannot view the customized login JSP pages. This happens when the customized JSP files are not restored or the `legacy` filesystem directory is not created.

To resolve this issue, see Maintaining Customized JSP Files for Identity Server in the NetIQ Access Manager Appliance 4.2 Installation and Upgrade Guide.

Troubleshooting the Administration Console

This section provides information about general troubleshooting issues found in the Administration Console:

- “Global Troubleshooting Options” on page 947
- “Diagnostic Configuration Export Utility” on page 950
- “Logging” on page 951
- “Restoring a Failed Secondary Console” on page 951
- “Converting a Secondary Access Manager Appliance into a Primary Appliance” on page 952
- “Repairing the Configuration Datastore” on page 956
- “Session Conflicts” on page 956
- “Unable to Log In to the Administration Console” on page 957
- “Exception Processing IdentityService_ServerPage.JSP” on page 957
- “Backup and Restore Failure Because of Special Characters in Passwords” on page 957
- “Unable to Install NMAS SAML Method” on page 958

- “Incorrect Audit Configuration” on page 958
- “Unable to Update the Access gateway Listening IP Address in the Administration Console Reverse Proxy” on page 959
- “During Access Manager Appliance Installation Any Error Message Should Not Display Successful Status” on page 960
- “Incorrect Health Is Reported on the Access Gateway” on page 960
- “Administration Console Does Not Refresh the Command Status Automatically” on page 960
- “SSL Communication with Weak Ciphers Fails” on page 960
- “Error: Tomcat did not stop in time. PID file was not removed” on page 961
- “An IP Address for the Other Known Device Manager List is Missing in the Troubleshooting Page” on page 961

Global Troubleshooting Options

The following options allow you to view the status of multiple devices and identify the devices that are not healthy.

- “Checking for Potential Configuration Problems” on page 947
- “Checking for Invalid Policies” on page 949
- “Checking for Version Conflicts” on page 949
- “Checking and Terminating User Sessions” on page 949
- “Checking for Invalid Policies” on page 950
- “Viewing System Alerts” on page 950

Checking for Potential Configuration Problems

If your Access Manager Appliance components are not behaving in the way you have configured them to run, you might want to check the system to see if any of the components have configuration or network problems.

1 In the Administration Console Dashboard, click **Troubleshooting** > **Configuration**.

2 All of the options should be empty, except the **Cached Access Gateway Configurations** option (see Step 4) and the **Current Access Gateway Configurations** option (see Step 5). If an option contains an entry, you need to clear it. Select the appropriate action from the following table:

Option	Description and Action
Device Pending with No Commands	If you have a device that remains in the pending state, even when all commands have successfully executed, that device appears in this list. Before deleting the device from this list, check its Command Status. If the device has any commands listed, select the commands, then delete them. Wait a few minutes. If the device remains in a pending state, return to this troubleshooting page. Find the device in the list, then click **Remove**. The Administration Console clears the pending state.

Option	Description and Action
Other Known Device Manager Servers	If a secondary Administration Console is in a non-reporting state, perhaps caused by hardware failure, its configuration needs to be removed from the primary Administration Console. As long as it is part of the configuration, other Access Manager devices try to contact it. If you cannot remove it by running the uninstall script on the secondary Administration Console, you can remove it by using this troubleshooting page. Click **Remove** next to the console that is in the non-reporting state. All references to the secondary Administration Console are removed from the configuration database.
Access Gateways with Corrupt Protected Resource Data	If you modify the configuration for a protected resource, update the Access Gateway with the changes, then review the configuration for the protected resource and the changes have not been applied, the configuration for the protected resource is corrupted. Click **Repair** next to the protected resource that has a corrupted configuration. You should then be able to modify its configuration, and when you update the Access Gateway, the changes should be applied and saved.
Access Gateways with Duplicate Protected Resource Data	After an upgrade, if you get errors related to invalid content for policy enforcement lists, you need to correct them. The invalid elements that do not have an associated resource data element are listed in this section. Click **Repair**.
Access Gateways with Protected Resources Referencing Nonexistent Policies	Protected resources have problems when policies are deleted before their references to the protected resources are removed. If you have protected resources in this condition, they are listed in this section. Click the **Repair** button to remove these references. Then verify that your protected resources have the correct policies enabled. Click **Access Gateways** > **Edit** > **[Name of Reverse Proxy]** > **[Name of Proxy Service]** > **Protected Resources**, then change to the **Policy View**.
Access Gateways with Invalid Alert Profile References	You can create XML validation errors on your Access Gateway Appliance if you start to create an alert profile (click **Access Gateways** > **Edit** > **Alerts** > **New**), but you do not finish the process. The incomplete alert profile does not appear in the configuration for the Access Gateway, so you cannot delete it. If such a profile exists, it appears in the **Access Gateways with Invalid Alert Profile References** list. Click **Remove**. You should then be able to modify its configuration, and when you update the Access Gateway, the changes should be applied and saved.
Devices with Corrupt Data Store Entries	If an empty value is written to an XML attribute, the device with this invalid configuration appears in this list. Click **Repair** to rewrite the invalid attribute values.

3 When you have finished repairing or deleting invalid Access Gateway configurations, click the **Access Gateways** link, then click **Update** > **OK**.

4 (Optional) Verify that all members of an Access Gateway cluster have the same configuration in cache:

4a Click **Troubleshooting** > **Configuration**.

4b Scroll to the **Cached Access Gateway Configuration** option.

4c Click **View** next to the cluster configuration or next to an individual Access Gateway.

This option allows you to view the Access Gateway configuration that is currently residing in browser cache. If the Access Gateway belongs to a cluster, you can view the cached configuration for the cluster as well as the cached configuration for each member. The + and - buttons allow you to expand and collapse individual configurations. The configuration is displayed in XML format

To search for particular configuration parameters, you need to copy and paste the text into a text editor.

5 (Conditional) After viewing the Access Gateway configuration (see Step 4) and discovering that an Access Gateway does not have the current configuration, select the Access Gateway in the **Current Access Gateway Configurations** section, then click **Re-push Current Configuration**.

Checking for Invalid Policies

The Policies page displays the policies that are in an unusable state because of configuration errors.

1 In the Administration Console Dashboard, click **Troubleshooting** > **Policies**.

If you have configured a policy without defining a valid rule for it, the policy appears in this list.

2 Select the policy, then click **Remove**.

Checking for Version Conflicts

The Version page displays all the installed components along with their currently running version. Use this page to verify that you have updated all components to the latest compatible versions. There are two steps to ensuring that your Access Manager Appliance components are running compatible versions:

To view the current version of all Access Manager Appliance devices:

1 In the Administration Console Dashboard, click **Troubleshooting**.

2 Click **Version**.

A list of the devices with their version information is displayed. If a device also has an Embedded Service Provider, the version of the Embedded Service Provider is also displayed.

Checking and Terminating User Sessions

The User Sessions page helps you to find users logged into your system and also helps to terminate their sessions if required. It displays the active user details for each Identity Server. You can search for a user with the user ID and terminate the session(s).

1 In the Administration Console Dashboard, click **Troubleshooting** > **User Sessions**.

2 Specify the user ID and click **Search**. If a match is found, it lists the IP address of the Identity Server and its sessions.

3 Click **Terminate Sessions** to terminate the sessions of the specific user.

NOTE: User details are fetched once per administration session. The last updated date is displayed. To refresh the data, click on **Refresh**.

For more information about user sessions, see "Terminating an Existing Authenticated User from the Identity Server" on page 996.

Checking for Invalid Policies

The Policies page displays the policies that are in an unusable state because of configuration errors.

1. In the Administration Console Dashboard, click **Troubleshooting** > **Policies**.

   If you have configured a policy without defining a valid rule for it, the policy appears in this list.
2. Select the policy, then click **Remove**.

Viewing System Alerts

The System Alerts page displays how many unacknowledged alerts have been generated for all the devices imported into this Administration Console.

1. In the Administration Console Dashboard, click **Alerts.**
2. To acknowledge and clear the alerts for a device, select the name of the server, then click **Acknowledge Alerts**.

The following columns display information about the alerts for each server.

Column	Description
Server Name	Specifies the name of the server receiving alerts. Click the server name to view more information about an alert before acknowledging it.
Severe	Indicates how many severe alerts have been sent to the server.
Warning	Indicates how many warning alerts have been sent to the server.
Informational	Indicates how many informational alerts have been sent to the server.

Diagnostic Configuration Export Utility

In the Administration Console, you can create a `.ldif` file that you can export for diagnostic purposes:

1. Log in as `root`.
2. **On Linux:** Change to the `/opt/novell/devman/bin` directory and run the following command:

   ```
   ./amdiagcfg.sh
   ```

   On Windows: Go to the `C:\Program Files (x86)\Novell\bin` directory and run the following command:

   ```
   ./amdiagcfg.bat
   ```
3. Specify the Access Manager administrator's password.
4. Confirm the password.
5. Specify the path where you want to store the diagnostic file.
6. Specify a name for the diagnostic file. The system adds .xml automatically as file extension.
7. Press Enter.

The Diagnostic Configuration Export utility is almost identical to the backup utility because it also creates a LDIF file with an addition of an XML Dump file. Passwords from the final LDIF file are removed by a program called Strippasswd.

Strippasswd removes occurrences of passwords in the LDIF file and replaces them with empty strings. If you look at the LDIF file, you will see that password strings are blank. You might see occurrences within the file or text that looks similar to password="String". These are not instances of passwords, but rather definitions that describe passwords as string types.

With every Access Manager release you must copy the corresponding XML style sheet file (XSL file) to the same directory where the XML file is located. This helps in displaying the information in correct format. The XSL file is located at `/opt/novell/devman/bin`.

The XML file along with XSL file or LDIF file (if required) can then be sent to NetIQ Support for help in diagnosing configuration problems.

Logging

You can troubleshoot by configuring component logging. In the Administration Console, click **Devices > Identity Server > Edit > Logging**.

For more information, see "Using Log Files for Troubleshooting" on page 1024.

Restoring a Failed Secondary Console

If a secondary console fails, you need to remove its configuration from the primary console before installing a new secondary console. If the failed console is part of the configuration, other Access Manager Appliance devices try to contact it.

1. On the primary console, click **Troubleshooting**.
2. In the **Other Known Device Manager Servers** section, click **Remove** next to the secondary console that is failed.
3. Remove traces of the secondary console from the configuration datastore:
   - **3a** In the NetIQ Access Manager menu bar, select **View Objects**.
   - **3b** In the Tree view, select **novell**.
   - **3c** Delete all objects that reference the failed secondary console.

     You should find the following types of objects:
     - SAS Service object with the hostname of the secondary console
     - An object that starts with the last octet of the IP address of the secondary console
     - DNS AG object with the hostname of the secondary console
     - DNS IP object with the hostname of the secondary console
     - SSL CertificateDNS with the hostname of the secondary console
     - SSL CertificateIP with the hostname of the secondary console
4. Install a new secondary console. For installation instructions, see "Installing Secondary Versions of Access Manager Appliance" on page 771.

Converting a Secondary Access Manager Appliance into a Primary Appliance

To convert a secondary Access Manager Appliance into a primary Access Manager Appliance, a recent backup of Access Manager Appliance must be available. For information about how to perform a backup, see “Backing Up the Access Manager Appliance Configuration” on page 926. A backup is necessary to restore the certificate authority (CA).

If the failed server holds a master replica of any partition, you must use `ndsrepair` to designate a new master replica on a different server in the replica list.

This conversion includes the following tasks:

- “Shutting Down the Primary Access Manager Appliance” on page 952
- “Changing the Master Replica” on page 952
- “Restoring CA Certificates” on page 953
- “Verifying the vcdn.conf File” on page 953
- “Deleting Objects from the eDirectory Configuration Store” on page 954
- “Performing Component-Specific Procedures” on page 954
- “Enabling Backup on the New Primary Appliance” on page 955

Shutting Down the Primary Access Manager Appliance

If your primary Access Manager Appliance is running, you must log in as the administrator and shut down the service.

Start YaST, click **System** > **System Services (Runlevel)**, then select to stop the ndsd service.

Changing the Master Replica

Changing the master replica to reside on the new primary Access Manager Appliance makes this Access Manager Appliance into the certificate authority for Access Manager. You need to first designate the replica on the new primary Access Manager Appliance as the master replica. Then you need to remove the old primary Access Manager Appliance from the replica ring.

- “Secondary Administration Console” on page 952

Secondary Administration Console

1. At the secondary Access Manager Appliance, log in as `root`.
2. Change to the `/opt/novell/eDirectory/bin` directory.
3. Run DSRepair with the following options:

   ```
   ./ndsrepair -P -Ad
   ```
4. Select the one available replica.
5. Select **Designate this server as the new master replica**.
6. Run `ndsrepair -P -Ad` again.
7. Select the one available replica.
8. Select **View replica ring**.
9. Select the name of the failed primary server.

10 Select **Remove this server from replica ring**.

11 Specify the DN of the admin user in leading dot notation. For example:

.admin.novell

12 Specify password.

13 Type `I Agree` when prompted.

14 Continue with "Restoring CA Certificates" on page 953.

Restoring CA Certificates

Perform the following steps on the machine that you are promoting to be a primary Appliance.

1 Copy your most recent Access Manager Appliance backup files to your new primary Access Manager Appliance.

2 Change to the backup `bin` directory:

`/opt/novell/devman/bin`

3 Verify the IP address in the backup file. The `IP_Address` parameter value should be the IP address of the new Primary Administration Console.

3a Open the backup file:

`defbkparm.sh`

3b Verify that the value in the IP_Address parameter is the IP address of your new primary console.

3c Save the file.

4 Run the certificate restore script:

`sh aminst-certs.sh`

5 When prompted, specify the administrator's password and location of the backup files.

6 Continue with "Verifying the vcdn.conf File" on page 953.

Verifying the vcdn.conf File

Verify whether the `vcdn.conf` file contains IP address of the new Administration Console. If it contains IP address of the failed primary Administration Console, replace it with the new IP address.

1 Change to the Appliance configuration directory:

`opt/novell/devman/share/conf`

2 Run the following command in the command line interface to restart Access Manager Appliance:

`/etc/init.d/novell-ac restart` OR `rcnovell-ac restart`

3 Continue with "Deleting Objects from the eDirectory Configuration Store" on page 954.

Deleting Objects from the eDirectory Configuration Store

Objects representing the failed primary Access Manager Appliance in the configuration store must be deleted.

1. Log in to the new Administration Console, then click **Access Gateways**.
2. If the failed primary Appliance's Access Gateway is the primary server (has the red icon next to it), then change the primary Access Gateway server.
   - **2a** Click **[Access Gateway cluster name]** > **Edit**.
   - **2b** Select a different primary Access Gateway > click **Ok** > click **Close**.

     Ignore any trust store related warnings.
   - **2c** Click **Update All**.

     Wait until the status becomes current for all except the failed primary Appliance.
3. Click **Troubleshooting**.
4. In the **Other Known Device Manager Servers** section, select the old primary Appliance, then click **Remove**.
5. Remove traces of the failed primary Access Manager Appliance from the configuration datastore:
   - **5a** In the NetIQ Access Manager menu bar, select **View Objects**.
   - **5b** In the Tree view, select **novell**.
   - **5c** Delete all objects that reference the failed primary Access Manager Appliance.

     You should find the following types of objects:
     - SAS Service object with the hostname of the failed primary console
     - An object that starts with the last octet of the IP address of the failed primary console
     - DNS AG object with the hostname of the failed primary console
     - DNS IP object with the hostname of the failed primary console
     - SSL CertificateDNS with the hostname of the failed primary console
     - SSL CertificateIP with the hostname of the failed primary console
     - NCP server object
6. Continue with “Performing Component-Specific Procedures” on page 954.

Performing Component-Specific Procedures

If you have installed the following components, perform the cleanup steps for the component:

- “Third Access Manager Appliance” on page 954
- “Access Gateway Services” on page 955

Third Access Manager Appliance

If you installed a third Appliance used for failover, you must manually perform the following steps on that server:

1. Open the `vcdn.conf` file.

   `/opt/novell/devman/share/conf`
2. In the file, look for the line that is similar to the following:

`<vcdnPrimaryAddress>10.1.1.1</vcdnPrimaryAddress>`

In this line, 10.1.1.1 represents the failed primary Appliance IP address.

3 Change this IP address to the IP address of the new primary Appliance.

4 Restart the Access Manager Appliance by entering the following command from the command line interface:

`/etc/init.d/novell-ac restart` OR `rcnovell-ac restart`

Access Gateway Services

For each Access Gateway Service imported into the Administration Console, edit the `settings.properties` file on the Access Gateway if the primary Administration Console was not configured as the Audit Server.

If the primary Administration Console was configured as an Audit Server, you must edit the `config.xml` file and the `settings.properties` file on the Access Gateway and edit the `CurrentConfig` and `WorkingConfig` XML documents in the configuration store on the new primary Administration Console.

1 At the Access Gateway Service, log in as the `root` or the `Administrator` user.

2 Shut down all Access Gateway Services.

`/etc/init.d/novell-appliance stop` OR `rcnovell-appliance stop`

3 (Conditional) If your audit server was on the primary Administration Console, edit the `config.xml` file:

3a Change to the directory and open the file.

`/opt/novell/nam/adminconsole/webapps/agm/WEB-INF/config/current`

3b `Find the NsureAuditSetting` entry.

In the `IPv4Address` field, change the IP address from the failed Administration Console to the new primary Appliance address.

3c Save and exit.

4 Edit the `settings.properties` file:

4a Change to the directory and open the file.

`/opt/novell/devman/jcc/conf`

4b Change the IP address in the `remotemgmtip` list from the IP address of the failed Appliance to the address of the new primary Appliance.

4c Save and exit.

5 At the Access Gateway Service, start all services by entering the following command:

`/etc/init.d/novell-appliance start` OR `rcnovell-appliance start`

6 (Conditional) Repeat this process for each Access Gateway Service that has been imported into the Administration Console.

Enabling Backup on the New Primary Appliance

1 On the new primary Appliance, change to the `/opt/novell/devman/bin` directory.

2 Open the `defbkparm.sh` file and find the following lines:

```
EDIR TREE=<tree_name>
EDIR CA=<CA name>
```

These lines contain values using the hostname of the Appliance you are on.

3 Modify these lines to use the hostname of the failed Appliance.

When you install the primary Appliance, the EDIR TREE parameter is set to the hostname of the server with _tree appended to it. The EDIR CA parameter is set to the hostname of the server with _tree CA appended to it.

If the failed Appliance had amlab as its hostname, you would change these lines to have the following values:

```
EDIR TREE="amlab_tree"
EDIR CA="amlab_tree CA"
```

4 Save your changes.

5 Take a backup from your new primary Appliance.

WARNING: After configuring the secondary Appliance to be the new primary Appliance and performing all the cleanup steps, you cannot restore an old backup from the primary Appliance.

Take a new backup as soon as your new primary Appliance is functional.

Repairing the Configuration Datastore

The configuration datastore is an embedded version of eDirectory 8.8. If it becomes corrupted, you can run DSRepair to fix it. Or, you can restore a recent backup. To restore a backup, see “Restoring the Access Manager Appliance Configuration” on page 927.

To run DSRepair:

1 In a browser, enter the following URL.

`http://<ip_address>:8028/nds`

Replace <*ip_address*> with the IP address of your Administration Console.

2 At the login prompt, enter the username and password of the admin user for the Administration Console.

The NDS iMonitor application is launched. For more information, see Accessing iMonitor (http://www.novell.com/documentation/edir88/edir88/data/a6l60f7.html).

3 In the **View** bar, select the **Repair** icon.

For more information about DSRepair, see the following:

- Click the **Help** icon.
- Using NdsRepair

Session Conflicts

Do not use two instances of the same browser to simultaneously access the same Administration Console. Browser sessions share settings, which can result in problems when you apply changes to configuration settings. However, you can use two different brands of browsers simultaneously, such as Internet Explorer and Firefox to avoid the session conflicts.

Unable to Log In to the Administration Console

If you experience problems logging in to the Administration Console, you might need to restart Tomcat.

1. Restart Tomcat by running this command:

   `/etc/init.d/novell-ac restart` OR `rcnovell-ac restart`

2. If this does not solve the problem, check the log file:

   `/opt/novell/nam/adminconsole/logs/catalina.out`

3. Check for the following error:

   ```
   Error Starting up core services.
   Application manager is Shutting down the Device Manager suite.
   Shutting down Device Manager suite.
   ```

4. If you see this error, check the status of eDirectory:

   4a Enter the following command:

   `/etc/init.d/ndsd status`

   If the status command returns nothing, start eDirectory manually.

   4b Enter the following command:

   `/etc/init.d/ndsd start`

   4c Restart Tomcat.

Exception Processing IdentityService_ServerPage.JSP

If you see the message `Exception processing IdentityService_ServerPage.jsp` on the Administration Console, it is an indication that the system has run out of available file handles. You need to use the command line to increase the ulimit value (`ulimit -n [new limit]`), which sets the number of open file descriptors allowed.

To set this value permanently, you can create the `/etc/profile.local` file with the ulimit value, such as:

```
ulimit -n 4096
```

You can make changes to `/etc/security/limits.conf` file with a line just to change the limit for a specific user. In this case: `novlwwuser`. Add the following line:

```
novlwww soft nofile [new limit]
```

Backup and Restore Failure Because of Special Characters in Passwords

Administration passwords with special characters such as dollar signs might cause the ambkup utility to fail. The ambkup utility creates a command line for the ICE utility, and the special characters might be interpreted by it. If you must use special characters, and this issue arises, modify the `defbkparm` file so that the special characters are escaped.

For example, if the administrator's password is mi$$le, then the field `DS_ADMIN_PWD` should be `mi\$\$le`.

This file is located in the following directory:

```
/opt/novell/devman/bin/defbkparm.sh
```

Unable to Install NMAS SAML Method

When you try to create an Identity Server cluster configuration with an eDirectory user store and with the **Install NMAS SAML method** option enabled and you have not installed the dependent packages, the following error message is displayed:

```
Warning: Failed to create SAML Affiliate Object
com.novell.security.japi.nmas.LoginMethodModel.getLsmWINNNTStatus()I
```

One of the installation requirements for the Administration Console is to install the compat and the libstdc++ packages.On SLES 11, the compat package contains the libstdc++ library. The Identity Server also requires the compat package. For more information about installing these packages, see TID 7006437.

Incorrect Audit Configuration

If the Audit Events from Access Gateway behind NAT are not seen in the Audit Server, do the following:

Click **Auditing** in the Administration Console and verify if values are provided for the **Server Listening IP Address, Server Public NAT IP Address**, and **Port Numbers** fields.

Scenario 1:

1. If the values are not provided for the **Server Listening IP Address, Server Public NAT IP Address**, and **Port Numbers** fields, enter the values, then click **Apply**.
2. If you change the existing values and click **Apply**, the following message is displayed:

   ```
   Step 1: Update all Access Gateways.
   Step 2: To update the configuration files in the Administration Console, see
   the context-sensitive help.
   Step 3: Reboot all servers to start using the new configuration.
   ```

3. Click **OK.**
4. Update the Access Gateway whose events are not seen.
5. Restart the Access Gateway.

Scenario 2:

1. If Server Listening IP Address, Server Public NAT IP Address and Port Numbers are valid and still have problems, repush the configuration.
2. Change the port number to some invalid port number, then click **Apply**.

   NOTE: Do not update or restart the Access Gateway as the message indicates.

3. Change the invalid port number again to the valid port number, then click **Apply**.

   The configuration is repushed and works successfully.
4. Update the Access Gateway whose events are not seen.
5. Restart the Access Gateway.

Unable to Update the Access gateway Listening IP Address in the Administration Console Reverse Proxy

The Administration Console fails to change the Access Gateway listening IP address of the Reverse Proxy. The health status of the Access Gateway on Administration Console displays failure to start the protected resource with old Listening IP address. However, when protected resource is viewed (**Devices** > **Access Gateways** > **Access Gateway or Access Gateway Cluster** > **Proxy**), the Administration Console displays the new IP Address has been selected as listening IP address of the Reverse Proxy.

To workaround this issue:

1. In the Administration Console, click **Devices** > **Access Gateways**.
   - **1a** Click the **Health** icon of the Access Gateway that has the problem.
   - **1b** Note the Reverse Proxies that have the problem.
2. In the Administration Console, click **Devices** > **Access Gateways**.
3. Click the **Edit** link for the cluster that has problem.
4. For each of the Reverse Proxies that have the problem, do the following:
   - **4a** Click **Reverse Proxy**.
   - **4b** Select the cluster member from the list.
   - **4c** Select the new IP address on which the proxy service will listen to.
   - **4d** Unselect the old IP address on which proxy service was listening to.
   - **4e** Click **OK**.
   - **4f** An alert is displayed as "Select at least one listening address for the service."
   - **4g** Click **OK**.
   - **4h** Again select the **Listening IP Address** check box.
   - **4i** Click **OK**.
5. If the update link is enabled, click on it. If not, do the following:
   - **5a** Click **Edit** for the cluster that has problem.
   - **5b** Click the **Proxy** name link.
   - **5c** Click **Proxy service name** in the **Proxy Service** list.
   - **5d** Enter the description.
   - **5e** Click **OK**.

     The **Update** link will be enabled.
   - **5f** Click **Update**.

After the device command status moves to Succeeded, verify the health status of the Access Gateway.

During Access Manager Appliance Installation Any Error Message Should Not Display Successful Status

Even after successful installation or upgrade of Access Gateway, the health shows failure in starting ESP. After an fresh import of Access Manager Appliance in the Administration Console, the Access Gateway Health displays *“ESP Failed to initialize : Unable to read <keystorefilelocation>”*. The keystore file can be Connector, Signing, Encryption or Truststore.

To workaround this issue:

1. On the Access Gateway, go to the <keystorefilelocation> location as specified in the health error message.
2. Delete the files indicated in the ESP error message.
3. In the Administration Console Dashboard, click **TroubleShooting > Certificates**.
4. Enable the device that has been deleted in the Access Manager Appliance and it needs to be re-pushed.
5. Click **Re-Push Certificate**.
6. Restart server provider of the Access Gateway.

Incorrect Health Is Reported on the Access Gateway

In the Administration Console, if the **Stop Service on Audit Server Failure** option is enabled, the Access Gateway services are stopped and show the Health status reports services as down when the Audit server is not functioning or reachable,.

If the **Stop Service on Audit Server Failure** option is disabled, the Access Gateway Service comes up but the related Health status still reports the services as being down.

To workaround this issue restart Tomcat.

Administration Console Does Not Refresh the Command Status Automatically

The automatic refresh feature to retrieve device health is disabled when total number of the Access Gateway devices imported to an Administration Console page is more than 20. This feature is disabled to prevent the performance overhead in getting the health of 20 or more devices simultaneously.

To workaround this issue an administrator can manually refresh the page to get the health status of the devices.

SSL Communication with Weak Ciphers Fails

Access Manger supports only the 128-bit SSL communication among the Administration Console, Identity Server, and browsers.

If you want to enable the weak ciphers (not recommended), see “Configuring the SSL Communication” on page 829.

Error: Tomcat did not stop in time. PID file was not removed

While stopping Tomcat for the Administration Console, Access Gateway, or Identity Server, you may get this error message:

```
Tomcat did not stop in time. PID file was not removed.
```

Ignore this message. Tomcat will be forcibly stopped if it does not stop normally.

An IP Address for the Other Known Device Manager List is Missing in the Troubleshooting Page

If the Administration Console is down, the IP address for that console is not seen. To bring up that Administration Console, follow these steps:

1. Run the `sntp -P no -r PRIMARY_IP` command.
2. Run the `/etc/init.d/novell-ac restart` OR `rcnovell-ac restart` command.

If the Administration Console is still not available, follow these steps:

1. Run the `/etc/init.d/ndsd restart` command.
2. Run the `/etc/init.d/novell-ac restart` OR `rcnovell-ac restart` command.

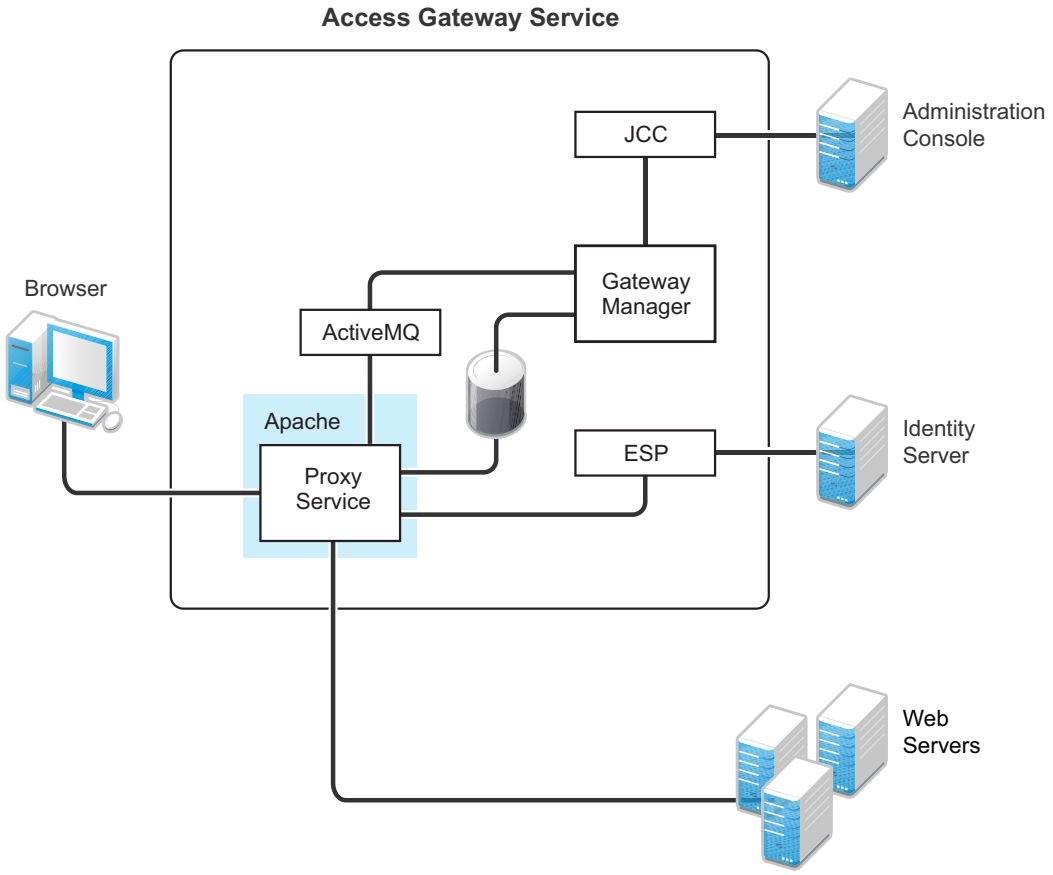
Troubleshooting the Access Gateway

- “Useful Troubleshooting Files” on page 962
- “Verifying That All Services Are Running” on page 964
- “Troubleshooting SSL Connection Issues” on page 966
- “Enabling Debug Mode and Core Dumps” on page 966
- “Useful Troubleshooting Tools for the Access Gateway Service” on page 968
- “Solving Apache Restart Issues” on page 968
- “Understanding the Authentication Process of the Access Gateway Service” on page 970
- “Enabling Caching of Audit Events for Apache Gateway Service” on page 977
- “Issue While Accelerating the Ajax Applications” on page 978
- “Accessing Lotus-iNotes through the Access Gateway Asks for Authentication” on page 978
- “Configuration Issues” on page 978
- “Cannot Inject a Photo into HTTP Headers” on page 978
- “Access Gateway Caching Issues” on page 979
- “Issues while Changing Management IP Address on an Access Gateway Appliance” on page 979
- “Issue while Adding the Access Gateway in a Cluster” on page 980

Useful Troubleshooting Files

The Access Gateway Service consists of two main modules, a Gateway Manager module that runs on top of Tomcat and a Proxy Service module that runs on top of Apache. Figure 26-1 illustrates these modules and the communication paths that the Access Gateway Service has with other devices.

Figure 26-1 *Access Gateway Service Modules*



Proxy Service: This component runs as an instance of Apache and is responsible for controlling access to the configured protected resources on the Web servers. Low-level errors are reported in the Apache logs. Some higher-level errors are also reported to the files in the `amlogging/logs` directory.

ESP: The Embedded Service Provider is responsible for handling all communications with the Identity Server and is responsible for the communication that verifies the authentication credentials of users. Log entries for this communication process, including errors, are logged in the `catalina.out` file and the `stdout.log` file.

ActiveMQ: This module is used for real-time communication between the Administration Console and the Proxy Service. Errors generated from the Gateway Manager to the ActiveMQ module are logged to the Tomcat logs. Errors generated from the Proxy Service to the ActiveMQ module are logged to the Apache error logs.

JCC: The Java Communication Controller is the interface to the Administration Console. It handles health, statistics, configuration updates, and purge cache requests from the Administration Console. It is also responsible for certificate management. Errors generated between the JCC module and the Gateway Manager are logged to the `ags_error.log` file. Errors generated between the Administration Console and the JCC module are logged to the `jcc-0.log.x` file

Gateway Manager: This module is responsible for handling communication from JCC to the Proxy Service. It also writes the configuration commands to the Apache configuration files and the Proxy Service configuration file on disk. Errors generated while performing these tasks are logged to the `ags_error.log` file.

For more information about these various log files, see the following:

- "Apache Logging Options for the Gateway Service" on page 963
- "The Access Gateway Service Log Files" on page 964

Apache Logging Options for the Gateway Service

The Proxy Service module of the Access Gateway Service is built on top of Apache as an Apache application. This module handles the browser requests for access to resources and is responsible for sending authorized requests to the Web servers. Entries for these events are logged to the Apache log files.

`/var/log/novell-apache2/`

For more information, see sections "Ignoring Some Standard Messages" on page 963 and "Managing Access Gateway Logs" on page 860.

Ignoring Some Standard Messages

Apache cannot detect the proper use of domain-based multi-homing with wildcard certificates, which allows multiple proxy services to share the same SSL port. If you create reverse proxy services that are configured for domain-based multi-homing with SSL, Apache considers this a possible port conflict and logs it as a warning in the `error.log` file.

The error messages look similar to the following:

```
[<time and date stamp>] [warn] Init: SSL server IP/port conflict:
dbmhnsnetid.dsm.cit.novell.com:443 (C:/Program
Files/Novell/apache/conf/vhosts.d/dbmhNS-NetID.conf:18) vs.
magwin1430external.dsm.cit.novell.com:443 (C:/Program
Files/Novell/apache/conf/vhosts.d/magMaster.conf:18)

[<time and date stamp>] [warn] Init: SSL server IP/port conflict:
magdbmheguide.dsm.cit.novell.com:443 (C:/Program
Files/Novell/apache/conf/vhosts.d/dbmhMagEguide.conf:18) vs.
magwin1430external.dsm.cit.novell.com:443 (C:/Program
Files/Novell/apache/conf/vhosts.d/magMaster.conf:18)
```

You can ignore these errors because the Access Gateway Service knows how to handle the traffic and send the packets to the correct proxy service.

For more information about Apache log files, see "Log Files" (http://httpd.apache.org/docs/2.2/logs.html).

Modifying the Logging Level for the Apache Logs

If the Apache error log file does not contain enough information, you can modify the log level and the types of messages written to the file.

WARNING: If you set the log level to debug, the size of the file can grow quickly, consume all available disk space, and crash the system. If you change the log level, you need to carefully monitor available disk space and the size of the error log file.

To modify what is written to the Apache error log file:

1. Change to the Apache configuration directory.

   `/etc/opt/novell/apache2/conf`

2. Open the `httpd.conf` file.
3. Find the `LogLevel` directive and set it to one of the following:

   `debug, info, notice, warn, error, crit, alert, emerg`

4. Save the file.
5. Restart Apache:

   `/etc/init.d/novell-apache2 restart` OR `rcnovell-apache2 restart`

6. (Optional) If you set the level to debug and the log file still does not supply enough information, see "Enabling Debug Mode and Core Dumps" on page 966.

The Access Gateway Service Log Files

See "Access Gateway Appliance and Access Gateway Service Logs" on page 870. You can gather these log files into a single zip file:

Verifying That All Services Are Running

1. Log in to the server as the `root` user.
2. Verify that the ActiveMQ service is running by entering the following command:

   ```
   ps -fea | grep novell-activemq | grep -v "grep"
   ```

   Lines similar to the following are displayed:

   ```
   107      18941     1  0 Apr01 ?        03:11:11 /opt/novell/java/bin/java -
   Xmx512M -Dorg.apache.activemq.UseDedicatedTaskRunner=true -
   Dcom.sun.management.jmxremote -Djavax.net.ssl.keyStorePassword=xxxxx -
   Djavax.net.ssl.trustStorePassword=xxxxx -Djavax.net.ssl.keyStore=/opt/novell/
   activemq/conf/broker.ks -Djavax.net.ssl.trustStore=/opt/novell/activemq/conf/
   broker.ts -Dactivemq.classpath=/opt/novell/activemq/conf; -Dactivemq.home=/
   opt/novell/activemq -Dactivemq.base=/opt/novell/activemq -jar /opt/novell/
   activemq/bin/run.jar start
   ```

3. Verify that one or more Apache proxy services are running by entering the following command:

   ```
   ps -ef | grep httpd
   ```

   Lines similar to the following are displayed:

```
root   2983 30290  0 12:53 pts/0  00:00:00 egrep httpd
root   3163     1  0 May12 ?      00:00:29 /opt/novell/apache2/sbin/httpd
wwwrun 3165  3163  0 May12 ?      00:01:00 /opt/novell/apache2/sbin/httpd
wwwrun 3184  3163  0 May12 ?      00:00:01 /opt/novell/apache2/sbin/httpd
wwwrun 3188  3163  0 May12 ?      00:00:01 /opt/novell/apache2/sbin/httpd
```

4 Verify that the user session cache service is running by entering the following command:

```
ps -ef | grep novell-agscd
```

Lines similar to the following are displayed:

```
root  3259 30290  0 12:56 pts/0    00:00:00 egrep novell-agscd
108   5525     1  0 May11 ?   00:00:00 /opt/novell/ag/bin/novell-agscd -d
108   5526  5525  0 May11 ?   00:00:09 /opt/novell/ag/bin/novell-agscd -d
```

5 Verify that the Tomcat service is running by entering the following command:

```
ps -ef | grep catalina.base
```

Lines similar to the following are displayed:

```
ps -eaf | grep catalina.base
novlwww  28764     1  0 Jul05 pts/0    00:02:05 /opt/novell/java/bin/java -Dnop
-server -Xmx2048m -Xms512m -Xss128k -Djava.library.path=/usr/lib64:/opt/
novell/eDirectory/lib64:/opt/novell/lib64 -
Dcom.novell.nam.common.util.DeploymentMode=MAGAppliance -
Dsun.net.client.defaultConnectTimeout=29000 -
Dsun.net.client.defaultReadTimeout=28000 -Dnids.freemem.threshold=10 -
Djavax.net.ssl.sessionCacheSize=10000 -
Dsun.net.http.allowRestrictedHeaders=true -Djava.awt.headless=true -
Djava.util.logging.manager=org.apache.juli.ClassLoaderLogManager -
Djava.endorsed.dirs=/var/opt/novell/tomcat7/endorsed -classpath /lib/
tools.jar:/var/opt/novell/tomcat7/bin/bootstrap.jar:/var/opt/novell/tomcat7/
bin/tomcat-juli.jar -Dcatalina.base=/opt/novell/nam/mag -Dcatalina.home=/var/
opt/novell/tomcat7 -Djava.io.tmpdir=/opt/novell/nam/mag/temp
org.apache.catalina.startup.Bootstrap -config /opt/novell/nam/mag/conf/
server.xml start
```

6 Verify that the JCC service is running by entering the following command:

```
ps -ef | grep /opt/novell/devman/jcc/conf/run.sh
```

Lines similar to the following are displayed:

```
root  3777 30290  0 13:03 pts/0 00:00:00 egrep /opt/novell/devman/jcc/
                                            conf/run.sh
root  5506     1  0 May11 ?      00:00:00 /bin/bash /opt/novell/devman/jcc/
                                            conf/run.sh
```

When you are familiar with the services, you can use the following command to display information about all the services:

```
ps -ef | egrep "novell-activemq|novell-agscd|/opt/novell/devman/jcc/conf/
run.sh|catalina.base|httpd"
```

7 If one or more services are not running, use the following commands to start the services:

`/etc/init.d/novell-jcc start` OR `rcnovell-jcc start`

`/etc/init.d/novell-apache2 start` OR `rcnovell-apache2 start`

`/etc/init.d/novell-agcsd start`

`/etc/init.d/novell-activemq start` OR `rcnovell-activemq start`

`/etc/init.d/novell-mag start` OR `rcnovell-mag start`

8 If a service does not start, view the log files to determine the cause. See the following:

- “Solving Apache Restart Issues” on page 968
- “The Access Gateway Service Log Files” on page 964

Troubleshooting SSL Connection Issues

SSL handshakes fail when there is a discrepancy between the cipher suites and cipher strengths used by the clients and the servers. If you enable SSL connections between the Access Gateway and the browser or between the Access Gateway and the Web servers, you need to make sure that both sides are configured to support the same cipher suites and cipher strengths. This is especially important if you enable the options to enforce 128-bit encryption (see “Configuring TCP Listen Options for Clients” on page 131).

The Access Gateway Service relies upon Apache to perform the SSL handshake, and Apache does not log the cause of SSL handshake failures, even when the log level is set to debug. To determine whether cipher strengths are the source of your problem, disable the options to enforce 128-bit encryption (see “Configuring TCP Listen Options for Clients” on page 131). If users are then able to authenticate, verify the cipher strengths, which are configured for the browsers and for the Web servers, are compatible with the Access Gateway.

Enabling Debug Mode and Core Dumps

If the log files are not generating enough information to identify the cause of a problem, you can run the Access Gateway Service in debug mode. You should not be running in debug mode except when you are trying to isolate a problem because of the following side effects:

- Debug mode causes the size of the log files to grow quickly. They can grow large enough to consume all available disk space and crash the system. When running in debug mode, you need to carefully monitor available disk space and the size of the log files.
- Debug mode opens additional ports. Anyone who has local access to the Access Gateway machine can see the information displayed in the following local URLs:

  ```
  http://127.0.0.1:8181/server-status
  http://127.0.0.1:8181/server-info
  ```

- Debug mode causes load and response times to slow.

Debug mode enables core dumps, X-Mag headers in LAN traces, and increases log levels by enabling advanced option in the Access Gateway configuration. For example LogLevel debug. This will set apache log level to debug in the `error_log` file.

You can generate core dumps in the following two ways:

1 Start `/etc/init.d/novell-apache2` in debug mode. When there is a crash, core file will be created as `/var/cache/novell-apache2/core`.

2 Without starting `novell-apache2` in debug mode, perform the following:

2a Set `ulimit -c unlimited` in `/etc/init.d/novell-apache2` startup script.

2b You can create the core directory under `/tmp`. Choose the file path based on the availability of disk space. Give the following command to create a directory in the Access Gateway component:

```
# mkdir -p /tmp/apache2-gdb-dump
```

2c Set permission as follows:

```
# chown novlwww:www /tmp/apache2-gdb-dump
# chmod 0777 /tmp/apache2-gdb-dump
```

2d Add the following advanced option in the Access Gateway configuration as follows:

```
CoreDumpDirectory /tmp/apache2-gdb-dump
```

2e Apply changes to the Access Gateway.

Whenever there is a crash, core file will be created as`/tmp/apache2-gdb-dump/core`.

For some crashes, the `/tmp/debug000.log` file is created. For more information about the log, see TID 7011804.

This section describes the following tasks:

- “Starting Apache in Debug Mode” on page 967
- “Examining the Debug Information” on page 967
- “Disabling Debug Mode” on page 967

Starting Apache in Debug Mode

Use the following commands to start debug mode:

`/etc/init.d/novell-apache2 stop` OR `rcnovell-apache2 stop`

`/etc/init.d/novell-apache2 start debug` OR `rcnovell-apache2 start debug`

Examining the Debug Information

1 Examine the Apache error log file or copy it so you can send it to NetIQ Technical Support:

`/var/log/novell-apache2`

2 View the information at the local URLs or copy the pages to send to NetIQ Support:

- http://127.0.0.1:8181/server-status

  This page displays debug information about caching, SSL, workers, and proxy information.

- http://127.0.0.1:8181/server-info

  This page displays module and configuration information.

3 If a crash occurred, examine the core dump file or copy it so you can send it to NetIQ Technical Support.

`/var/cache/novell-apache2`

Disabling Debug Mode

Use the following commands to disable debug mode:

`/etc/init.d/novell-apache2 stop` OR `rcnovell-apache2 stop`

`/etc/init.d/novell-apache2 start nodebug` OR `rcnovell-apache2 start nodebug`

Useful Troubleshooting Tools for the Access Gateway Service

Table 26-1 describes some of the tools available in the Administration Console for solving potential problems:

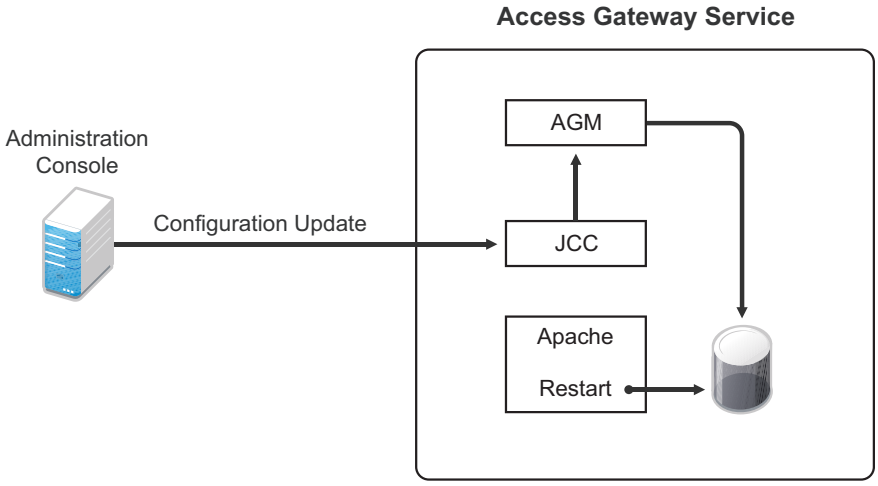
Table 26-1 *Useful Tools*

Tool	Description
Re-push Current Configuration	If you have an Access Gateway that does not seem to be using the current configuration, you can use the Administration Console to push the current configuration to the Access Gateway. Click **Troubleshooting**. In the **Current Access Gateway Configuration** section, select an Access Gateway, then click **Re-push Current Configuration**.
Health icon	In the Administration Console, click the **Health** icon to view details about the health of the Access Gateway. For more information, see "Monitoring the Health of an Access Gateway" on page 904.

Solving Apache Restart Issues

When you make configuration changes and update the Access Gateway, the Administration Console uses the JCC channel to send the configuration changes to the Access Gateway. Figure 26-2 illustrates this flow.

Figure 26-2 *Sending Configuration Updates to the Access Gateway*



JCC sends the configuration changes to the Access Gateway Manager (AGM), which writes the Apache configuration to disk. Apache is sent a restart command, which causes Apache to read the new configuration, then Apache validates the configuration.

- If the configuration is valid, Apache starts.
- If the configuration is invalid, Apache fails to start.

If Apache fails to start after a configuration change, roll back to the previous configuration. Restore a backup if you have one, or use the Administration Console to manually remove the modifications that have caused the problem. If this does not solve the problem, try the following:

- “Removing Any Advanced Configuration Settings” on page 969
- “Viewing the Logged Apache Errors” on page 969
- “Viewing the Errors as Apache Generates Them” on page 969
- “The ActiveMQ Module Fails to Start” on page 970

Removing Any Advanced Configuration Settings

Apache fails to start when it discovers a syntax error in any of the advanced options.

1. Click **Devices > Edit > Advanced Options**.
2. To reset all options to their default values, delete all options from the text box.
3. Click **OK**.

   When you return to the Advanced Options page, all options are set to their default values.
4. Click **[Name of Reverse Proxy] > [Name of Proxy Service] > Advanced Options**.
5. To reset all options to their default value, delete all options from the text box.
6. Click **OK**.

   When you return to the Advanced Options page, all options are set to their default values.
7. Repeat these steps for each proxy service that has advanced options configured.
8. Update the Access Gateway.

Viewing the Logged Apache Errors

Apache generates and logs errors when it fails to start. A summary is displayed on the health page.

1. In the Administration Console, click **Devices > Access Gateways > Health**.

   The page displays a summary of the problem from the Apache error log file. For the Access Gateway Service, information from the `rcnovell-apache2.out` file might also be displayed.
2. To view the entire contents of the Apache error log file, open a terminal window to the Access Gateway.
3. Change to the following directory and open the Apache error log file.

   `/var/log/novell-apache2`
4. iew the contents of the `rcnovell-apache2.out` file.
5. If you still do not have enough information to solve the configuration problem, continue with “Viewing the Errors as Apache Generates Them” on page 969.

Viewing the Errors as Apache Generates Them

Apache allows only a few errors to be sent to log files. To view all the errors, use the following procedure to display the errors in a terminal window.

1. Copy the `config.xml` file in the `current` directory to a temporary location. The Access Gateway allows only one XML file to reside in the `current` directory.

   `/opt/novell/nam/mag/webapps/agm/WEB-INF/config/current`

2 Copy the XML file from the `pending` directory to the current directory and rename it `config.xml`.

   The file in the `pending` directory has a long numeric name.

3 Change the ownership of the file from `root` to `novlwww:novlwww`.

4 Use one of the following commands to restart Tomcat:

   `/etc/init.d/novell-mag restart` OR `rcnovell-mag restart`

5 Restart Apache by using the following command:

   `/etc/init.d/novell-apache2 restart` OR `rcnovell-apache2 restart`

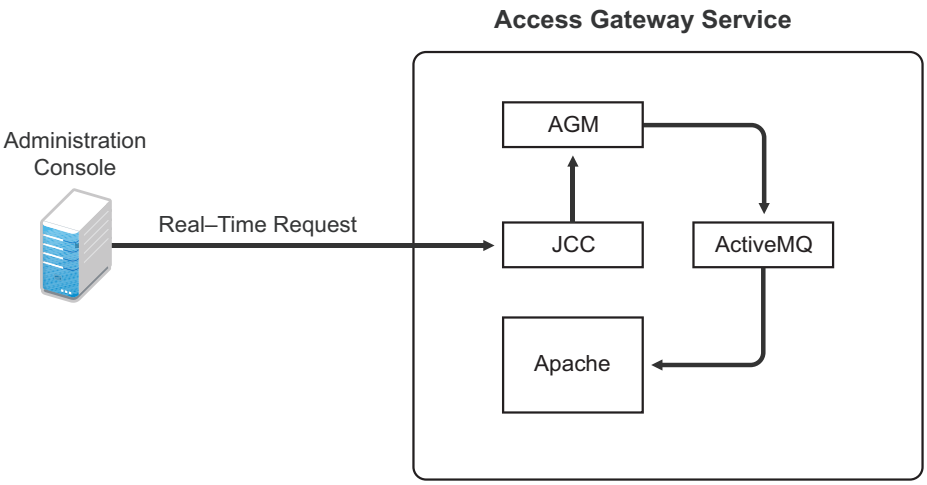
   Apache uses the terminal window to write the errors it discovers as it tries to process the `config.xml` file.

6 At the Administration Console, fix the configuration problems, then update the Access Gateway.

The ActiveMQ Module Fails to Start

The Active MQ module is used for real-time communication between the Administration Console and the Access Gateway Service. Real-time communication is needed for commands such as purging cache, gathering statistics, and updating health. Figure 26-3 illustrates this communication flow.

Figure 26-3 *Real-Time Communication*



When the ActiveMQ module fails to start, you cannot apply any configuration changes, and the Access Gateway does not set a listener for the configured port.

To start the module, it must be able to resolve the listening IP address to a DNS name. To install an Access Gateway Service, the machine must have a DNS name and the IP address must resolve to this name.

Understanding the Authentication Process of the Access Gateway Service

When a user requests access to a protected resource, the request can be in one of the following states:

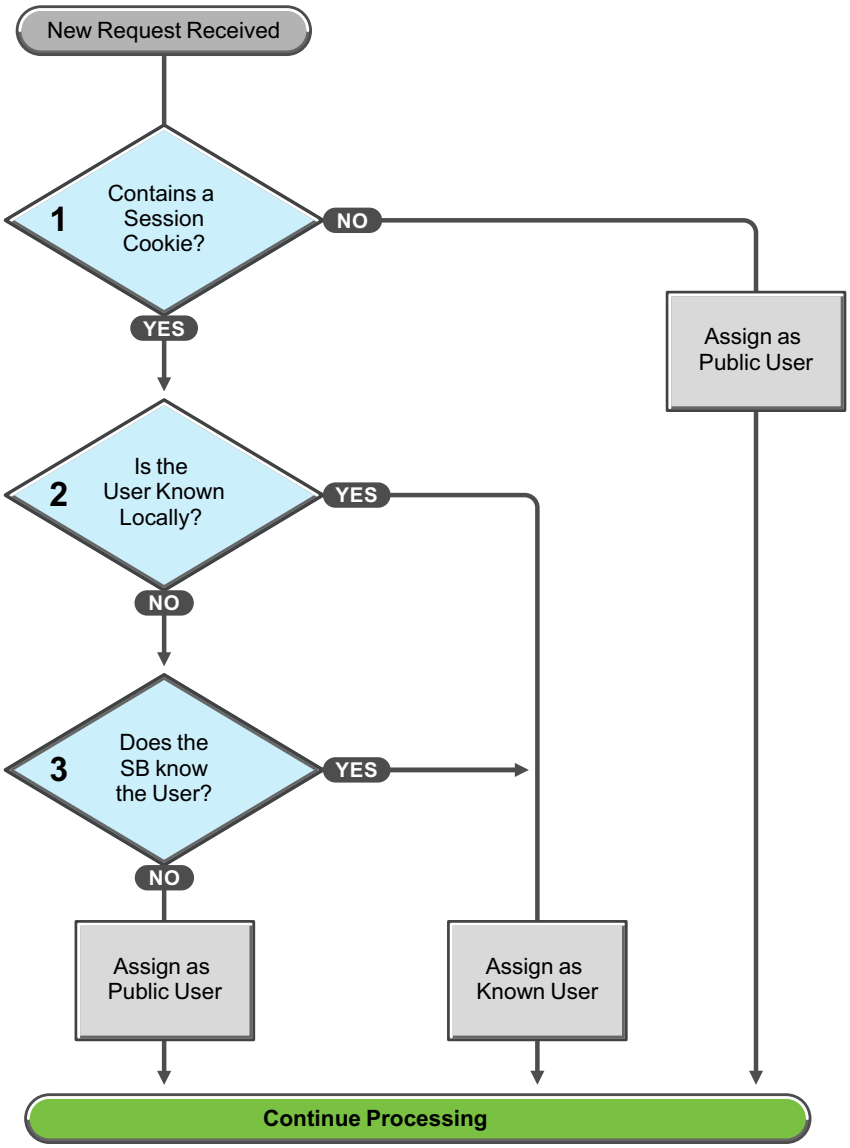
- No session or cookie is established, because this is the user's first request.
- The user's session is a public session because only public resources have been accessed.

- A session is established, the user is authenticated, and the requested resource is from the same cookie domain and uses the same contract.
- A session is established, the user is authenticated, and the requested resource is from the same cookie domain but uses a different contract or the contract has expired.
- A session is established, the user is authenticated, but the request does not' have a session cookie because the resource is on a different cookie domain.
- A session no longer exists or does not' exist on the proxy servicing the request.

The Access Gateway Service must handle these conditions and others as it determines whether it needs to forward a login request to the Embedded Service Provider or use the user's existing authentication credentials. The following flow charts take you through this process:

- Figure 26-4, "Identifying the Requester," on page 972
- Figure 26-5, "Determining the Type of Request," on page 973
- Figure 26-6, "Determining the Protection Type Assigned to the Resource," on page 975
- Figure 26-7, "Evaluating the Cookie Domain," on page 976

Figure 26-4 *Identifying the Requester*



These first steps determine whether the Access Gateway knows the user that has submitted the request. In decision point 1, the Access Gateway checks for a session cookie in the request.

- If the request contains a session cookie, the session cookie needs to be validated. Processing continues with the task in decision point 2.
- If the request does not contain a session cookie, the user is unknown and is assigned as a public user. The Access Gateway continues processing with the tasks outlined in Figure 26-5 on page 973.

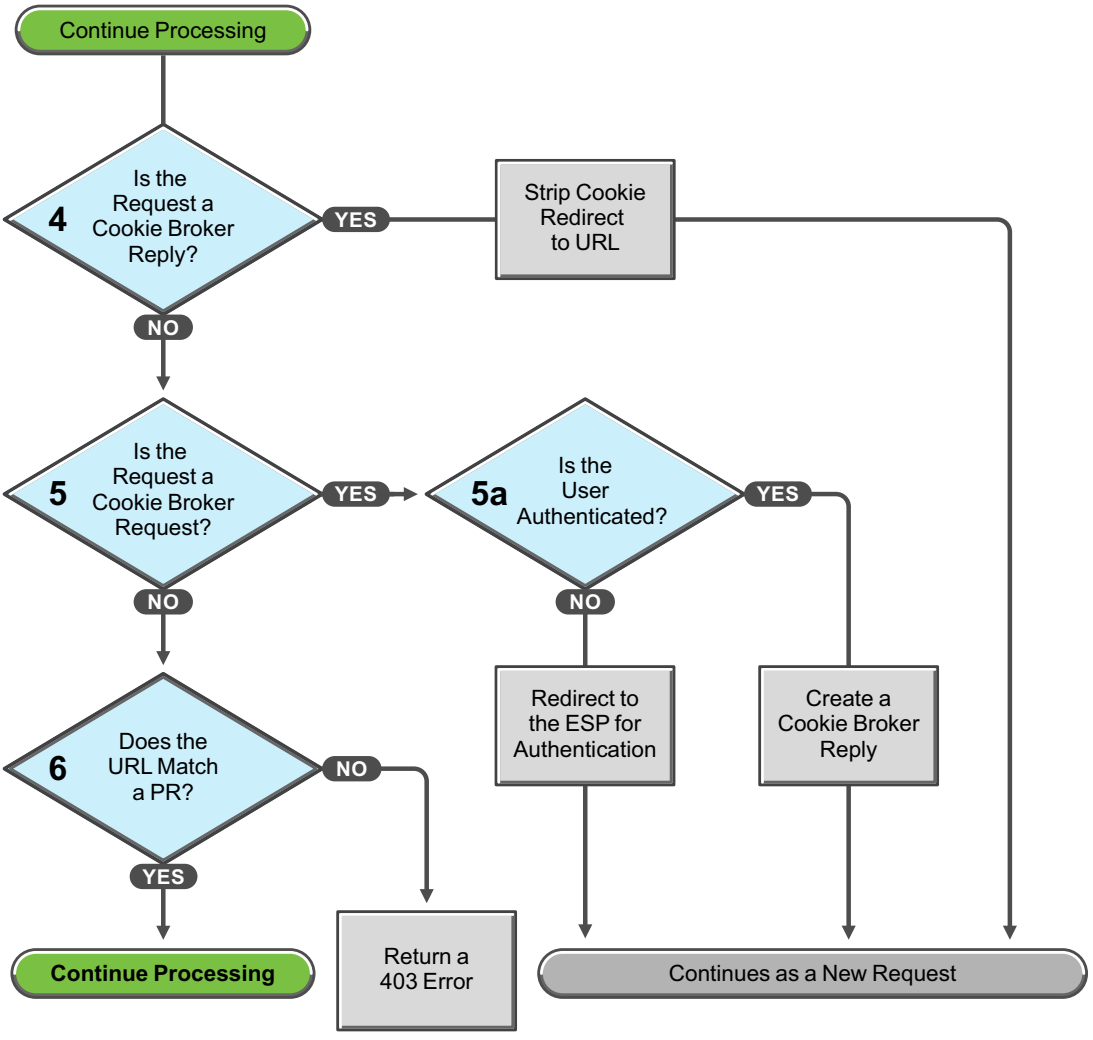
When the request contains a session cookie, the Access Gateway checks its local user store for a user that matches the session cookie. Each Access Gateway in the cluster maintains its own list of known users.

- If the session cookie matches one of the locally known users, the user is assigned that identity. The Access Gateway continues with the tasks outlined in Figure 26-5 on page 973.
- If the session cookie does not' match one of the locally known users, the Access Gateway needs to know if one of the other Access Gateways in the cluster knows the user. Processing continues with the task in decision point 3.

The Access Gateway queries the session broker to see if one of the other Access Gateways in the cluster knows this user.

- If a match is found, the user is assigned that identity. The Access Gateway continues with tasks outlined in Figure 26-5 on page 973.
- If a match is not found, the user is unknown and is assigned as a public user. The Access Gateway continues with the tasks outlined in Figure 26-5 on page 973.

Figure 26-5 *Determining the Type of Request*



The Access Gateway examines the request to determine what type of request it is.

If the request is a cookie broker reply, the Access Gateway strips the cookie from the URL and redirects the request to the URL. The redirect is handled as a new request, and this new request flows to the task in decision point 6, where the URL is examined.

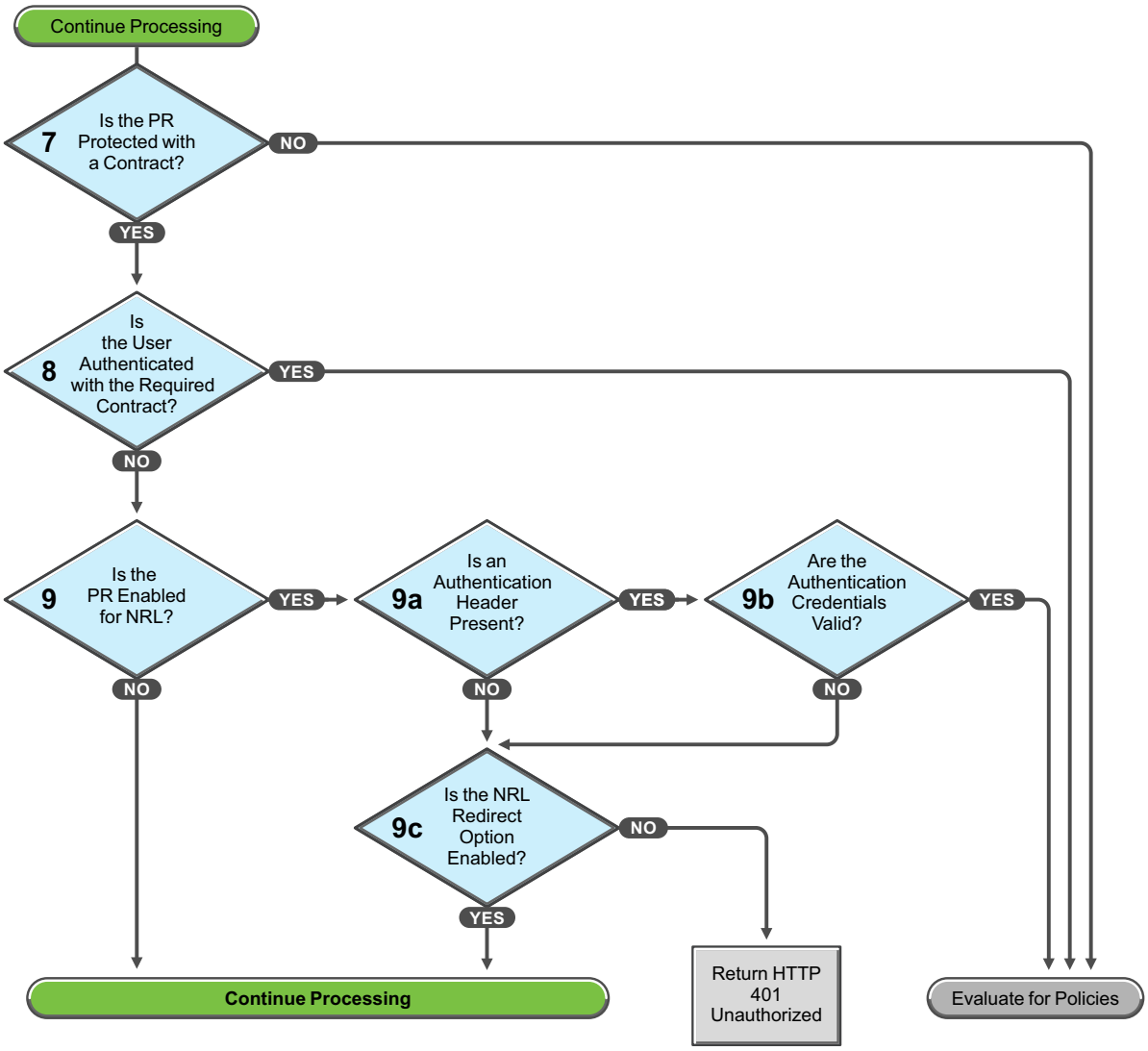
If the request is not' a cookie broker reply, the Access Gateway examines the request to see if it is a cookie broker request. If it is a cookie broker request, the Access Gateway determines whether the user is authenticated with the contract required by the protected resource.

- If the user is authenticated, the Access Gateway creates a cookie broker reply. This reply is handled as a new request, and flows to the task in decision point 4.
- If the user is not authenticated, the request is redirected to the Embedded Service Provider (ESP). The ESP interacts with the Identity Server to authenticate the user. The Identity Server, the ESP, and the reverse proxy all maintain authentication information. The ESP returns a new request, which flows to the task in decision point 6, where the URL is examined.

If the URL does not match a URL of a protected resource (PR), the Access Gateway returns an HTTP 403 error to the user.

If the URL in the request matches a URL of a protected resource, the Access Gateway needs to examine the protection type assigned to the resource. The Access Gateway continues with the tasks outlined in Figure 26-6 on page 975.

Figure 26-6 *Determining the Protection Type Assigned to the Resource*



You configure a protected resource as a public resource when an authentication procedure/contract is not assigned to the protected resource. In decision point 7, the Access Gateway checks to see if a contract has been assigned to the protected resource.

- If the protected resource has not been assigned a contract, the Access Gateway is finished with its authentication checks and continues with policy evaluation.
- If the protected resource has been assigned a contract, the Access Gateway continues with the task in decision point 8.

For a user to gain access to a resource protected by a contract, the user must have authenticated with that contract, or if the contract is configured for it, the user can authenticate with another contract as long as the contract is of a equal or higher level.

- If the user is authenticated with the required contract, the Access Gateway is finished with its authentication checks and continues with policy evaluation.
- If the user is not authenticated with the required contract, the Access Gateway continues with the task in decision point 9.

Before the user is prompted for credentials, the Access Gateway needs to know whether the protected resource has been enabled for non-redirected login (NRL).

- If the resource has not been configured for non-redirected login, the Access Gateway continues with the tasks outlined in Figure 26-7 on page 976.
- If the resource has been configured for non-redirected login, the Access Gateway needs to examine the request for an authentication header and determine whether the header is valid. Processing continues with the tasks outlined in decision points 9a, 9b, and 9c.

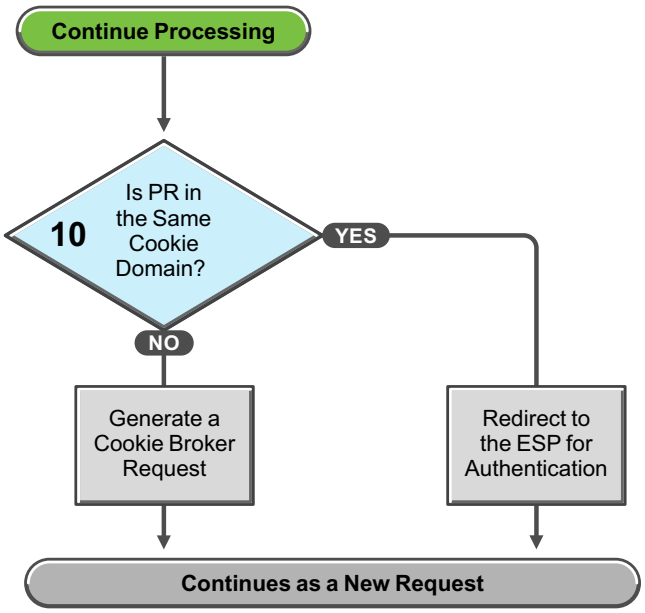
If the request does not contain an authentication header, the Access Gateway needs to determine how non-redirected login has been configured. On the Authentication Procedure configuration page, you can select to enable the **Redirect to Identity Server When No Authentication Header Is Provided** option.

- If this option is enabled, the Access Gateway continues with the tasks outlined in Figure 26-7 on page 976.
- If this option is disabled, the Access Gateway returns an HTTP 401 unauthorized message.

If the request does contain an authentication header, the Access Gateway must verify that the credentials are valid.

- If the authentication credentials are valid, the Access Gateway is finished with its authentication checks and continues with evaluating the protected resource for policies.
- If the authentication credentials are not valid, the process is the same as if the request did not contain an authentication header and continues with the task in decision point 9c.

Figure 26-7 *Evaluating the Cookie Domain*



If you have configured your Access Gateway to use multiple domain-based proxy services, you can configure them to share the same cookie domain (domains of `development.novell.com` and `support.novell.com` can share the cookie domain of `novell.com`) or configure them so that they cannot share a cookie domain (domains of `a.slc.com` and `b.provo.com` cannot share a cookie domain).

When the Access Gateway reaches the task in decision point 10, it has determined that the protected resource requires a contract and that user is not authenticated with that contract.

- If the protected resource is in the same cookie domain, the Access Gateway redirects the request to the Embedded Service Provider (ESP). The ESP interacts with the Identity Server to authenticate the user. The ESP returns a new request, which flows to the task in decision point 6, where the URL is examined.
- If the protected resource is in a different cookie domain, the Access Gateway generates a cookie broker request. This new request flows to the task in decision point 5.

Enabling Caching of Audit Events for Apache Gateway Service

Access Gateway sends audit events to the Audit Server through Platform Agent. You can enable the caching of events on Platform Agent. If the Audit Server goes down, Platform Agent will cache all audit events triggered during that period and send these to Audit Server when the Audit Server is up. Enabling caching prevents any loss of event data.

NOTE: Platform Agent and Novell Audit are no longer supported. A new Access Manager 4.2 installation no longer installs Platform Agent and Novell Audit for auditing. If you upgrade from an older version of Access Manager to 4.2, Platform Agent is still available. It is recommended to use Syslog for auditing.

---

1. Open the `log4j.xml.base` file.

   `/etc/opt/novell/amlogging/config/log4j.xml.base`

2. Look for Audit Server entry. By default the **EnableCaching** value is set to false. The xml entry for Audit server looks as below:

```
<appender name="AMAuditNSureAuditAppender"
class="com.novell.nacm.logging.audit.AMNSureAuditAppender">
        <param name="AppendMode" value="DIRECT"/>
        <param name="ErrorHandling" value="DISCARD"/>
        <param name="CertificatePath"
value="/etc/opt/novell/amlogging/certs/amnacert.pem"/>
        <param name="PrivateKeyPath"
value="/etc/opt/novell/amlogging/certs/amnapkey.pem"/>
        <param name="EnableCaching" value="false"/>
        <param name="ServerCheckInterval" value="4"/>

        <filter class="com.novell.nacm.logging.audit.AMNSureAuditFilter">
        </filter>
    </appender>
```

3. Modify the **EnableCaching** value from **false** to **true**.
4. In the Administration Console, click **Devices** > **Access Gateways** > **Edit** > **Auditing**. If any of the events are enabled, then disable all the events by deselecting them. Click **OK** twice. On the Access Gateways page, click **Update**.

Issue While Accelerating the Ajax Applications

If you are accelerating an Ajax application that cannot handle redirect and uses an authentication contract of 5 or 10 min, then increase the contract time out. Ensure that your Ajax application refreshes at an interval of 2 or 5 min. As a best practice, ensure that the Ajax application refresh interval is less than 2/3 of the contract time out.

Accessing Lotus-iNotes through the Access Gateway Asks for Authentication

This issue in not related to Access Manager. You need to configure authentication in Lotus-iNotes.

For more information about configuring Lotus-iNotes, see section 2.1 Authentication (http://www.redbooks.ibm.com/redbooks/pdfs/sg246518.pdf) in the *iNotes Web Access Deployment and Administration guide.*

Configuration Issues

If you get pending configuration issues when you apply changes on the device, one of the reasons could be that the soft link for the `certs` folder does not exist.

Enter the following command to check if the soft link exists for the `certs` folder:

```
ls -ltrh /opt/novell/apache2/
```

The following output is displayed:

```
lrwxrwxrwx 1 root root   34 2012-03-09 19:43 certs -> /etc/opt/novell/apache2/conf/
certs
```

If the soft link does not exist, perform the following steps:

1. Enter the following command:

   ```
   ln -sf /etc/opt/novell/apache2/conf/certs opt/novell/apache2/conf/certs
   ```

2. Click **Troubleshooting** > **Certificates**.
3. Select the store that is reporting errors, then click **Re-push certificates**. You can select multiple stores at the same time.
4. (Optional) To verify that the re-push of the certificates was successful, click **Security** > **Command**

Cannot Inject a Photo into HTTP Headers

You can use the jpegPhoto LDAP attribute to store your photo in JPEG format. This LDAP attribute is not injecting the image into a custom HTTP header and returns a 400 Bad Request error.

Edit the `index.php` file and add the following line:

```
<img src="....Base64 value got from
customheader......."/>
```

Access Gateway Caching Issues

If you have caching issues with inodes, disk space, and cache corruption in the Access Gateway, use Apache htcacheclean tool which is used to keep the size of mod_disk_cache's storage within a certain limit. This tool can run either manually or in daemon mode. When running in daemon mode, it sleeps in the background and checks the cache directories at regular intervals for cached content to be removed.

The htcacheclean utility tool is located at:

On Linux: /opt/novell/apache2/sbin

The default cache location is:

On Linux: `/var/cache/novell-apache2`

Example: To clear 1024 MBytes of cache, run the following command:

On Linux: `./htcacheclean -v -t -p/var/cache/novell-apache2 -l1024M`

For more information about Apache htcacheclean tool, see htcacheclean - Clean up the disk cache (https://httpd.apache.org/docs/2.2/programs/htcacheclean.html).

Issues while Changing Management IP Address on an Access Gateway Appliance

If the Access Gateway Appliance has two NICs, a public and a private NIC, it is unable to change the Management IP address of the Access Gateway Appliance to a new value. The Administration Console connects to the changed IP address, but also tries to connect to the old IP address. The following procedure allows you to change the IP address manually.

1. Stop the AG service /etc/init.d/novell-mag stop
2. Stop the JCC service by using the `/etc/init.d/novell-jcc stop` command.
3. Change the IP address in `/opt/novell/devman/jcc/conf/settings.properties` file.
4. Change the IP address in `/opt/novell/nam/mag/webapps/agm/WEB-INF/config/current/config.xml` file.
5. Change the IP address in `/etc/opt/novell/apache2/conf/listen.conf` file.
6. Using YaST, change the network IP address.
7. In the Administration Console Edir, edit and change the IP address in the following attributes:
   - For a specific Access Gateway device entry:
     1. romaAGDeviceXMLDoc of ou=ag-xxxxxx, ou=AppliancesContainer, ou=Partition, ou=PartitionsContainer, ou=VCDN_Root, ou=accessManagerContainer, o=novell
     2. romaAGDeviceSAXMLDoc of ou=ag-xxxxxx, ou=AppliancesContainer, ou=Partition, ou=PartitionsContainer, ou=VCDN_Root, ou=accessManagerContainer, o=novell

3. romaAGConfigurationXMLDoc of ou=WorkingConfig, ou=ag-xxxx, ou=AppliancesContainer, ou=Partition, ou=PartitionsContainer, ou=VCDN_Root, ou=accessManagerContainer, o=novell
4. romaAGConfigurationXMLDoc of ou=CurrentConfig ou=ag-xxxx, ou=AppliancesContainer, ou=Partition, ou=PartitionsContainer, ou=VCDN_Root, ou=accessManagerContainer, o=novell

- For the specific Access Gateway (esp)-identity server entry:
  1. romaIDPDeviceSAXMLDoc of ou=idp-esp-xxxxxx, ou=AppliancesContainer, ou=Partition, ou=PartitionsContainer, ou=VCDN_Root, ou=accessManagerContainer, o=novell
  2. romaIDPDeviceXMLDoc of ou=idp-esp-xxxxxx, ou=AppliancesContainer, ou=Partition, ou=PartitionsContainer, ou=VCDN_Root, ou=accessManagerContainer, o=novell
- For other Access Gateway device entry (if they are in a cluster):
  1. romaAGConfigurationXMLDoc of ou=WorkingConfig, ou=ag-yyyy, ou=AppliancesContainer, ou=Partition, ou=PartitionsContainer, ou=VCDN_Root, ou=accessManagerContainer, o=novell
  2. romaAGConfigurationXMLDoc of ou=CurrentConfig, ou=ag-yyyy, ou=AppliancesContainer, ou=Partition, ou=PartitionsContainer, ou=VCDN_Root, ou=accessManagerContainer, o=novell
- For tmp folder entry:
  1. romaAGConfigurationXMLDoc of ou=CurrentConfig, ou=tmp_zzz, ou=AppliancesContainer, ou=Partition, ou=PartitionsContainer, ou=VCDN_Root, ou=accessManagerContainer, o=novell
  2. romaAGConfigurationXMLDoc of ou=WorkingConfig, ou=tmp_zzz, ou=AppliancesContainer, ou=Partition, ou=PartitionsContainer, ou=VCDN_Root, ou=accessManagerContainer, o=novell

8 Start the Access Gateway service by using the `/etc/init.d/novell-mag start` command.

9 Start the JCC service by using the `/etc/init.d/novell-jcc start` command.

10 Restart the Administration Console, Identity Server and other Access Gateways in cluster.

Issue while Adding the Access Gateway in a Cluster

You may get the following error while adding the Access Gateway in a cluster:

Unable to read keystore: /opt/novell/devman/jcc/certs/esp/4C06F0AE2EFAED18/signing.keystore

To workaround this issue:

1 Click **Troubleshooting** > **Certificates**.

2 Select the store that is reporting errors, then click **Re-push certificates**.

You can select multiple stores at the same time.

3 (Optional) To verify that the re-push of the certificates was successful, click **Security** > **Command Status**.

Troubleshooting Identity Server and Authentication

This section provides information about the following topics:

- “Useful Networking Tools for Linux Identity Server” on page 982
- “Troubleshooting 100101043 and 100101044 Liberty Metadata Load Errors” on page 982
- “Authentication Issues” on page 989
- “After Setting Up the User Store to Use SecretStore, Users Report 500 Errors” on page 991
- “When Multiple Browser Logout Option Is Enabled, User Is Not Getting Logged Out from Different Sessions” on page 991
- “After Consuming a SAML Response, the Browser Is redirected to an Incorrect URL” on page 992
- “Configuring SAML 1.1 Identity Provider Without Specifying Port in the Login URL Field” on page 992
- “Attributes Are Not Available Through Form Fill When OIOSAML Is Enabled” on page 992
- “Issue in Importing Metadata While Configuring Identity Provider or Service Provider Using Metadata URL” on page 992
- “Metadata Mentions Triple Des As Encryption Method” on page 993
- “Issue in Accessing Protected Resources with External Identity Provider When Both Providers Use Same Cookie Domain” on page 993
- “SAML Intersite Transfer URL Setup Does Not Work for Non-brokered Setups after Enabling SP Brokering” on page 993
- “Orphaned Identity Objects” on page 993
- “Users cannot Log In to Identity Provider When They Access Protected Resources With Any Contract Assigned” on page 994
- “Attribute Query from OIOSAML.SP Java Service Provider Fails with Null Pointer” on page 994
- “Disabling the Certificate Revocation List Checking” on page 994
- “Step up Authentication for the Identity Server Initiated SSO to External Provider Does not Work Unless It has a Matching Local Contract” on page 995
- “Metadata Cannot be Retrieved from the URL” on page 995
- “Requesting the Authentication to a Service Provider Fails” on page 995
- “SAML 2.0 POST Compression Failure Does not Throw a Specific Error Code” on page 995
- “SAML 1.1 Service Provider Re-requests for Authentication” on page 995
- “The Identity Server Statistics Logs Do Not Get Written In Less Than One Minute” on page 996
- “No Error Message Is Written in the Log File When an Expired Certificate Is Used for the X509 Authentication” on page 996
- “Terminating an Existing Authenticated User from the Identity Server” on page 996
- “X.509 Authentication Lists the Entire List of Certificates Imported to the Browser” on page 997
- “Clustered Nodes Looping Due to JGroup Issues” on page 997
- “Authentication With Aliases Fails” on page 998
- “nidp/app Does Not Redirect to nidp/portal after Authentication” on page 998
- “Login to Office 365 Fails when WS-Trust MEX Metadata Is Larger than 65 KB” on page 998
- “Unsafe Server Certificate Change in SSL/TLS Renegotiations Is Not Allowed” on page 999
- “Viewing Request and Response Headers of All Protocols in a Log File” on page 999

For information about Identity Server logging, see “Configuring Logging for Identity Server” on page 852 and “Configuring Session-Based Logging” on page 854.

Useful Networking Tools for Linux Identity Server

You can use the following tools (Linux and open source) to troubleshoot network problems:

- **netstat:** Displays information related to open ports on your server. Lets you view listeners and various IP addresses, such as the TCP output state.
- **iptables:** Allows you to change the default ports (8080 and 8443) to the standard ports (80 and 443) for HTTP traffic.
- **netcat:** A networking utility that reads and writes data across network connections, using the TCP/IP protocol. Netcat is useful for checking connectivity with the user store.
- **ldapsearch:** An LDAP search tool useful for the Administration Console and Identity Server. For example, you can generate an LDAP search/bind matching what the Identity Server sends, to confirm whether an issue is with the Identity Server JAR files.
- **tcpdump:** A command line tool for monitoring network traffic. Captures and displays packet headers and matches them against a set of criteria.
- **LDAP Browser/Editor:** Lets you export configuration information to a file, and to confirm that Access Manager objects and attribute values are valid in an AccessManagerContainer. A number of open source versions are available from the Internet.

Troubleshooting 100101043 and 100101044 Liberty Metadata Load Errors

The Identity Server is the identity provider for other Access Manager components. Access Gatewayshas Embedded Service Providers. When a device is imported into the Administration Console and an Identity Server configuration is selected for them, a trusted relationship is established with the Identity Server by using test certificates. When you change these certificates or change from using HTTP to HTTPS, you need to ensure that the trusted relationship is reestablished. Metadata is used for establishing trusted relationships.

The metadata exchanged between service providers and identity providers contains public key certificates, key descriptors for message signing, a URL for the SSO service, a URL for the SLO (single logout) service, and so on. With Access Manager, this metadata is accessible on both the Identity Server and the Embedded Service Provider of the device. Errors are generated when either the identity provider could not load the service provider’s metadata (100101043), or the service provider could not load the metadata of the identity provider (100101044).

If users are receiving either of these errors when they attempt to log in, verify the following:

- “The Metadata” on page 983
- “DNS Name Resolution” on page 984
- “Certificates in the Required Trust Stores” on page 985

If these steps do not solve your problem, try the following:

- “Enabling Debug Logging” on page 986
- “Testing Whether the Provider Can Access the Metadata” on page 988

- "Manually Creating Any Auto-Generated Certificates" on page 989
- For information about metadata validation process and the flow of events that occur when accessing a protected resource on the Access Gateway, see "Troubleshooting 100101043 and 100101044 Errors in Access Manager" (http://www.novell.com/coolsolutions/appnote/19456.html).

The Metadata

If you change the base URL of the Identity Provider, all service providers, including Embedded Service Providers, need to be updated so that they use the new metadata:

- "Embedded Service Provider Metadata" on page 983
- "Service Provider Metadata" on page 983

Embedded Service Provider Metadata

If you change the base URL of the Identity Provider, all Access Manager devices that have an Embedded Service Provider need to be updated so that new metadata is imported. To force a re-import of the metadata, you need to configure the device so it does not' have a trusted relationship with the Identity Server, update the device, reconfigure the device for a trusted relationship, then update the device. The following steps explain how to force the Access Gateway to re-import the metadata of the Identity Server.

1. In the Administration Console, click **Devices** > **Access Gateways** > **Edit** > **Reverse Proxies/Authentication**.
2. Select **None** for the **Identity Server Cluster** option, click **OK** twice, then update the Access Gateway.
3. Click **Edit** > **Reverse Proxies/Authentication**.
4. Select an Identity Server configuration for the **Identity Server Cluster** option, click **OK** twice, then update the Access Gateway.

Service Provider Metadata

If you have set up federation with another provider over the Liberty, SAML 1.1, SAML 2.0, or WS Federation protocol and you change the base URL of the Identity Server, you need to update the provider with the new metadata to reestablish the trusted relationship. If the provider is another Identity Server, follow the procedure below to update the metadata; otherwise, follow the provider's procedures.

1. In the Administration Console of the provider, click **Devices** > **Identity Servers** > **Edit** > **[Protocol]** > **[Provider]** > **Metadata**.
2. Click **Reimport**.
3. Follow the steps in the wizard.

   For more information, see "Managing Metadata" on page 158.

DNS Name Resolution

When the service provider tries to access the metadata on the identity provider, it sends the request to the hostname defined in the base URL configuration of the Identity Server. The base URL in the Identity Server configuration is used to build all the metadata end points.

To view the metadata of the Identity Server with a DNS name of `idpcluster.lab.novell.com`, enter the following URL:

```
https://idpcluster.lab.novell.com:8443/nidp/idff/metadata
```

Scan through the document and notice the multiple references to `https://idpcluster.lab.novell.com/...` You should see lines similar to the following:

```
<md:SoapEndpoint>
   https://idpcluster.lab.novell.com:8443/nidp/idff/soap
</md:SoapEndpoint>

<md:SingleLogoutServiceURL>
   https://idpcluster.lab.novell.com:8443/nidp/idff/slo
</md:SingleLogoutServiceURL>

<md:SingleLogoutServiceReturnURL>
   https://idpcluster.lab.novell.com:8443/nidp/idff/slo_return
</md:SingleLogoutServiceReturnURL>
```

The Embedded Service Provider of the Access Gateway must be able to resolve the `idpcluster.lab.novell.com` hostname of the Identity Server. To test that it is resolvable, send a `ping` command with the hostname of the Identity Server. For example, from the Access Gateway:

```
ping idpcluster.lab.novell.com
```

The same is true for the Identity Server. It must be able to resolve the hostname of the Access Gateway. To discover the URL for the Access Gateway metadata:

1. In the Administration Console, click **Devices** > **Access Gateways** > **Edit** > **Reverse Proxy/Authentication**.
2. View the **Embedded Service Provider** section.

   The URL of the metadata is displayed in this section.

To view the metadata, enter the displayed URL. Scan through the document and notice the multiple references to the hostname of the Access Gateway. You should see lines similar to the following. In these lines, the hostname is ag1.provo.novell.com.

```
<md:SoapEndpoint>
   http://ag1.provo.novell.com:80/nesp/idff/spsoap
</md:SoapEndpoint>

<md:SingleLogoutServiceURL>
   http://ag1.provo.novell.com:80/nesp/idff/spslo
</md:SingleLogoutServiceURL>

<md:SingleLogoutServiceReturnURL>
   http://ag1.provo.novell.com:80/nesp/idff/spslo_return
</md:SingleLogoutServiceReturnURL>
```

To test that the Identity Server can resolve the hostname of the Access Gateway, send a `ping` command with the hostname of the Access Gateway. For example, from the Identity Server:

```
ping ag1.provo.novell.com
```

To view sample log entries that are logged when a DNS name cannot be resolved, see “The Embedded Service Provider Cannot Resolve the Base URL of the Identity Server” on page 987.

Certificates in the Required Trust Stores

Ensure that the issuers of the Identity Server and Embedded Service Provider certificates are added to the appropriate trusted root containers.

When the server certificates are sent from the identity provider to the service provider client, and from the service provider to the identity provider client, the client needs to be able to validate the certificates. Part of the validation process is to confirm that the server certificate has been signed by a trusted source. By default, well known external trusted certificates are bundled with Access Manager. You can view this list here: **Administration Console** > **Security** > `Certificates` > **External Trusted Roots**. If the issuer of server certificate is not present in the External Trusted Root list, the import the issuers of the server certificate (intermediate and trusted roots) into the correct trusted root stores:

- The intermediate and trusted roots of the Embedded Service Provider certificate must be imported into the NIDP Trust Store.
- The intermediate and trusted roots of the Identity Server certificate must be imported into the ESP Trust Store.

For more information, see “Importing a Signed Certificate” on page 805.

If you use certificates generated by the Administration Console CA, the trusted root certificate is the same for the Identity Server and the Embedded Service Provider. If you are using external certificates, the trusted root certificate might not be the same, and there might be intermediate certificates that need to be imported.

To verify the trusted root certificates:

1. In the Administration Console, click **Security** > **Certificates**.
2. Determine the issuer of the Identity Server certificate and the Embedded Service Provider certificate:
   - **2a** Click the name of the Identity Server certificate, note the name of the Issuer, then click **Close**.
   - **2b** Click the name of the Embedded Service Provider certificate of the Access Gateway, note the name of the Issuer, then click **Close**.
3. To verify the trusted root for the Identity Server, click **Devices**> **Identity Servers** > **Edit** > **Security** > **NIDP Trust Store**.
4. In the **Trusted Roots** section, scan for a certificate subject that matches the issuer of the Embedded Service Provider certificate, then click its name.
   - If the Issuer has the same name as the Subject name, then this certificate is the root certificate.
   - If the Issuer has a different name than the Subject name, the certificate is an intermediate certificate in the chain. Click **Close**, and ensure that another certificate in the trust store is the root certificate. If it isn’t there, you need to import it and any other intermediate certificates between the one you have and the root certificate.
5. To verify the trusted root for the Embedded Service Provider, click **Devices** > **Access Gateways** > **Edit** >**Service Provider Certificates**> **Trusted Roots**.

6 In the Trusted Roots section, scan for a certificate subject that matches the issuer of the Identity Server certificate, then click its name.

- If the Issuer has the same name as the Subject name, then this certificate is the root certificate.
- If the Issuer has a different name than the Subject name, the certificate is an intermediate certificate in the chain. Click **Close**, and ensure that another certificate in the trust store is the root certificate. If it isn't there, you need to import it and any other intermediate certificates between the one you have and the root certificate.

7 (Optional) If you have clustered your Identity Servers and Access Gateways and you are concerned that not all members of the cluster are using the correct trusted root certificates, you can re-push the certificates to the cluster members.

7a Click **Troubleshooting** > **Certificates**.

7b Select the Trust Store of your Identity Servers and Access Gateways, then click **Re-push certificates**.

7c Update the Identity Severs and Access Gateways.

7d Check the command status of each device to ensure that the certificate was pushed to the device. From the Identity Servers page or the Access Gateways page, click the **Commands** link.

To view sample log entries that are logged to the `catalina.out` file when a trusted root certificate is missing, see "Trusted Roots Are Not Imported into the Appropriate Trusted Root Containers" on page 987.

Enabling Debug Logging

You can enable Identity Server logging to dump more verbose Liberty information to the `catalina.out` file on both the Identity Server and the Embedded Service Provider of the Access Gateway.

1 In the Administration Console, click **Devices** > **Identity Servers** > **Edit** > **Logging**.

2 Select **Enabled** for **File Logging** and **Echo to Console**.

3 In the **Component File Logger Levels** section, set **Application** and **Liberty** to a **debug** level.

4 Click **OK**, update the Identity Server, then update the Access Gateway.

5 After enabling and applying the changes, duplicate the issue once more to add specific details to the log file for the issue.

If the error is the 100101044 error, look at the log file on the Embedded Service Provider for the error code

If the error is the 100101043 error, look at the log file on the Identity Server for the error code.

On Linux, look at the `catalina.out` file, and on Windows, look at the `stdout.log` file.

6 (Conditional) To view the log files from the Administration Console, click **Auditing** > **General Logging**, then select the file and download it.

7 (Conditional) To view the log files on the device, change to the `log` directory.

- On Linux, change to the `/var/opt/novell/nam/logs`/idp directory.
- On Windows Server 2012, change to the `/Program Files (x86)/Novell/Tomcat/logs` directory.

Below are a few typical entries illustrating the most common problems. They are from the `catalina.out` file of the Embedded Service Provider:

- "The Embedded Service Provider Cannot Resolve the Base URL of the Identity Server" on page 987
- "Trusted Roots Are Not Imported into the Appropriate Trusted Root Containers" on page 987
- "The Server Certificate Has an Invalid Subject Name" on page 988

The Embedded Service Provider Cannot Resolve the Base URL of the Identity Server

When the Embedded Service Provider cannot resolve the DNS name of the Identity Server, the metadata cannot be loaded and a hostname error is logged. In the following entries, the Embedded Service Provider cannot resolve the `idpcluster.lab.novell.com` name of the Identity Server.

```
<amLogEntry> 2009-08-06T16:24:56Z INFO NIDS Application: AM#500105024:
AMDEVICEID#esp-09C720981EEE4EB4: AMAUTHID#2CA1168DF7343A42C7879
E707C51A03C: ESP is requesting metadata from IDP https://
idpcluster.lab.novell.com/nidp/idff/metadata </amLogEntry>

<amLogEntry> 2009-08-06T16:24:56Z SEVERE NIDS IDFF: AM#100106001:
AMDEVICEID#esp-09C720981EEE4EB4: Unable to load metadata for Embedded
Service Provider: https://idpcluster.lab.novell.com/nidp/idff/
metadata, error: AM#300101046: AMDEVICEID#esp-09C720981EEE4EB4::   Attempted to
connect to a url with an unresolvable host name
</amLogEntry>

<amLogEntry> 2009-08-06T16:24:56Z INFO NIDS Application: AM#500105039:
AMDEVICEID#esp-09C720981EEE4EB4: AMAUTHID#2CA1168DF7343A42C7879
E707C51A03C: Error on session id 2CA1168DF7343A42C7879E707C51A03C,
error 100101044-esp-09C720981EEE4EB4, Unable to authenticate.
AM#100101044: AMDEVICEID#esp-09C720981EEE4EB4:: Embedded Provider
failed to load Identity Provider metadata </amLogEntry>
```

Trusted Roots Are Not Imported into the Appropriate Trusted Root Containers

When the trusted roots are not imported into the appropriate trusted root containers, a certificate exception is thrown and an untrusted certificate message is logged. In the following log entries, the Embedded Service Provider is requesting metadata from the Identity Server, but the Embedded Service Provider does not trust the Identity Server certificate because the trusted root of the issuer of the Identity Server certificate is not in the Embedded Service Provider's trusted root container.

```
<amLogEntry> 2009-08-05T16:07:53Z INFO NIDS Application: AM#500105024:
AMDEVICEID#esp-09C720981EEE4EB4: AMAUTHID#D983B08C28D35221D13 9D33E5324F98F: ESP
is requesting metadata from IDP https://idpcluster.lab.novell.com/nidp/idff/
metadata </amLogEntry>

<amLogEntry> 2009-08-05T16:07:53Z SEVERE NIDS IDFF: AM#100106001: AMDEVICEID#esp-
09C720981EEE4EB4: Unable to load metadata for Embedded ServiceProvider: https://
idpcluster.lab.novell.com/nidp/idff/metadata, error:
java.security.cert.CertificateException: Untrusted Certificate-  chain </
amLogEntry>

<amLogEntry> 2009-08-05T16:07:53Z INFO NIDS Application: AM#500105039:
AMDEVICEID#esp-09C720981EEE4EB4: AMAUTHID#D983 B08C28D35221D139 D33E5324F98F:
Error on session id D983B08C28D35221D139D33E5324F98F, error 100101044-esp-
09C720981EEE4EB4, Unable to authenticate. AM#100101044: AMDEVICEID#esp-
09C720981EEE4EB4:: Embedded Provider failed to load Identity Provider metadata </
amLogEntry>
```

The Server Certificate Has an Invalid Subject Name

When the certificate has an invalid subject name, the handshake fails. In the log entries below, the Embedded Service Provider is requesting metadata from the Identity Server. The server certificate name does not match, so the Embedded Service Provider is unable to authenticate and get the metadata necessary to establish the trusted relationship.

```
<amLogEntry> 2009-07-05T16:07:53Z INFO NIDS Application: AM#500105024:
AMDEVICEID#esp-09C720981EEE4EB4: AMAUTHID#D983B08C28D35221D139D33 E5324F98F: ESP
is requesting metadata from IDP
https://idpcluster.lab.novell.com/nidp/idff/metadata </amLogEntry>

<amLogEntry> 2009-07-05T16:07:53Z SEVERE NIDS IDFF: AM#100106001: AMDEVICEID#esp-
09C720981EEE4EB4: Unable to load metadata for Embedded Service Provider: https://
idpcluster.lab.novell.com/nidp/idff/metadata, error: Received fatal alert:
handshake_failure </amLogEntry>

<amLogEntry> 2009-07-05T16:07:53Z INFO NIDS Application: AM#500105039:
AMDEVICEID#esp-09C720981EEE4EB4: AMAUTHID#D983B08C28D35221D139D33 E5324F98F: Error
on session id D983B08C28D35221D139D33E5324F98F, error 100101044-esp-09C720981EEE
4EB4, Unable to authenticate. AM#100101044: AMDEVICEID#esp-09C720981EEE4EB4: :
Embedded Provider failed to load Identity Provider
metadata </amLogEntry>
```

Testing Whether the Provider Can Access the Metadata

To test whether the metadata is available for download, enter the metadata URL of the identity provider and service provider. If the DNS name of the identity provider is `idpcluster.lab.novell.com`, open a browser at the Identity Server and enter the following URL:

```
https://idpcluster.lab.novell.com:8443/nidp/idff/metadata
```

Open a browser on the Access Gateway Service, then enter the same URL.

Because the Access Gateway Appliance does not have a graphical interface, you need to use the `curl` command to test whether the Access Gateway Appliance can access the metadata of the Identity Server. If the DNS name of the identity provider is `idpcluster.lab.novell.com`, enter the following command from the Access Gateway machine:

```
curl -k https://idpcluster.lab.novell.com:8443/nidp/idff/metadata
```

To test whether the Identity Server can access the metadata URL of the Access Gateway, open a browser on the Identity Server machine. If the published DNS name of service provider is `www.aleris.net`, enter the following URL:

```
https://www.aleris.net/nesp/idff/metadata
```

Manually Creating Any Auto-Generated Certificates

Occasionally, there are issues where the subject name was auto-generated and the entire configuration appears to be correct, but the 100101044/100101043 error is still reported. Delete the auto-generated certificate and manually re-create the server certificate, making sure that it is added to the relevant devices and stores.

Authentication Issues

This section discusses the following issues that occur during authentication:

- "Authentication Classes and Duplicate Common Names" on page 989
- "General Authentication Troubleshooting Tips" on page 989
- "Slow Authentication" on page 990
- "Federation Errors" on page 990
- "Mutual Authentication Troubleshooting Tips" on page 990
- "Browser Hangs in an Authentication Redirect" on page 991

Authentication Classes and Duplicate Common Names

If users have the same common name and exist in different containers under the same authentication search base, one or more attributes in addition to the common name must be configured for authentication to uniquely identify the user. You can set up an authentication class to handle duplicate common names.

1. Select either the name/password or secure name/password class.
2. Add two properties to the class:
   - **Query:** The value of the Query attribute needs to be a valid LDAP query string. Field names from the JSP login form can be used in the LDAP query string as variables for LDAP attribute values. The variables must be enclosed between two % characters. For example, (&(objectclass=person)(cn=%Ecom_User_ID%)(mail=%Ecom_Email%)) queries for an object of type person that contained a common name equal to the Ecom_User_ID field from the specified JSP form and mail equal to the Ecom_Email field from the same JSP form.
   - **JSP:** The JSP property value needs to be the name of a new `.jsp` file that includes all the needed fields for the Query property. The value of this attribute does not include the `.jsp` extension of the file. For example, if you create a new `.jsp` file named `login2.jsp`, the value of the JSP property is `login2`.

     For more information about creating custom login pages that prompt for more than username and password, see "Customizing the Identity Server Login Page" on page 193.

General Authentication Troubleshooting Tips

- Use LAN traces to check requests, responses, and interpacket delay times.
- In the user store logs, confirm that the request arrived. Check for internal errors.

- If you have created an admin user for the user store, ensure that the user has sufficient rights to find the users in the specified the search contexts. For more information about the required rights, see “Configuring an Admin User for the User Store” on page 276.
- Check the user store health and replica layout. See TID 3066352.
- Ensure that the user exists in the user store and that the user’s context is defined as a search context.
- Ensure that the Liberty protocol is enabled if you have configured Access Manager devices to use the Identity Server for authentication (click **Identity Servers** > **Edit** > **General Configuration**).
- Check the properties of the class and method. For example, the search format on the properties must match what you’ve defined on a custom login page. You might be asking for a name/password login, but the method specifies e-mail login criteria.
- Enable authentication logging options (click **Identity Servers > Edit > Logging**).
- Ensure that the authentication contract matches the base URL scheme. For example, check to see if SSL is used across all components.

Slow Authentication

The following configuration problems can cause slow authentication:

- If authentication is taking up to a minute per user, verify that your DNS server has been enabled for reverse lookups. The JNDI module in the Identity Server sends out a request to resolve the IP address of the LDAP server to a DNS name. If your DNS server is not enabled for reverse lookups, it takes 10 seconds for this request to fail before the Identity Server can continue with the authentication request.
- If your user store resides on SUSE Linux Enterprise Server 10, which installs with a firewall, you must open TCP 524. For more information about the ports that must be open when a firewall separates the user store from other Access Manager components, see Setting Up Firewalls in the NetIQ Access Manager Appliance 4.2 Installation and Upgrade Guide.
- If your LDAP user store is large, ensure that the search contexts are as specific as possible to avoid searching the entire tree for a user.

Federation Errors

- Most errors that occur during federation occur because of time synchronization problems between servers. Ensure that all of your servers involved with federation have their time synchronized within one minute.
- When the user denies consent to federate after clicking a Liberty link and logging in at the identity provider, the system displays an error page. The user should acknowledge that federation consent was denied and return to the service provider login page. This is the expected behavior when a user denies consent.

Mutual Authentication Troubleshooting Tips

- LAN traces:
  - Check the SSL handshake and look at trusted root list that was returned.

- The client certificate issuer must be in the identity provider certificate store and be applied to all the devices in a cluster.
- Ensure that the user exists and meets the authentication criteria. As the user store administrator, you can search for a subject name (or certificate mapping attributes defined) to locate a matching user.

- Enable the **Show Certificate Errors** option on the Attributes page for the X.509 authentication class. (Click **Identity Servers > Servers > Edit > Local > Classes > [x.509] > Properties**.) Enabling this option provides detailed error messages on the login browser, rather than generic messages.
- Ensure that the certificate subject name matches the user you log in with, if you are chaining methods.
- Use NTRadPing to test installations.
- Verify that the correct UDP port 1812 is specified.
- Verify that the RADIUS server can accept requests from the Identity Server. This might require the NAS-IP-Address attribute along with credentials.
- Verify that the user exists in the user store if multiple methods are added to a contract.
- Verify that user authentication works independent of Access Manager.
- Verify that the NMAS server is local and no tree walks are occurring across the directory.
- Ensure that the NMAS_LOGIN_SEQUENCE property is defined correctly.

Browser Hangs in an Authentication Redirect

If the browser hangs when the user attempts to authenticate at an identity provider, determine whether a new authentication contract was created and set as the default contract on the Identity Server. If this is the case and you have an Access Gateway resource set to accept any contract from the identity provider, you should navigate to the **Overview** tab for the protected resource and specify **Any** again in the **Contract** drop-down menu. Then click **OK**, then update the Access Gateway.

After Setting Up the User Store to Use SecretStore, Users Report 500 Errors

If your eDirectory user store is running on SLES 11 SP1 64-bit (or a higher version) on x86-64 hardware, you can install the NMAS SAML method for SecretStore from the Administration Console, but the eDirectory server is missing the required support libraries.

When users try to enter values for SecretStore entries in a form, they receive the following message:

```
Status: 500 Internal Server Error, Description: Datastore Error
```

To correct the problem, you need to install the missing libraries on your eDirectory server. For instructions, see TID 7006437 (http://www.novell.com/support/viewContent.do?externalId=7006437&sliceId=1).

When Multiple Browser Logout Option Is Enabled, User Is Not Getting Logged Out from Different Sessions

Allow multiple browser session logout option in the Identity Server cluster specifies whether a user with more than one session to the server is presented with an option to log out of all sessions. If you do not select this option, only the current session can be logged out. If you deselect this option in

instances where multiple users log in as guests, then when one user logs out, none of the other guests are logged out. When you enable this option, you must also restart any Embedded Service Providers that use this Identity Server configuration and for logout URL you have to configure it as `/nidp/app/logout`.

After Consuming a SAML Response, the Browser Is redirected to an Incorrect URL

After consuming a SAML response, 302 redirect to RelayState URL is sent to an Incorrect URL. Check whether the relay state is URL encoded. To fix this issue, add the following entry in the `web.xml` file:

```
(/opt/novell/nids/lib/webapp/WEB-INF/web.xml)<context-param> <param-
name>decodeRelayStateParam</param-name> <param-value>true</param-value></context-
param>
```

Configuring SAML 1.1 Identity Provider Without Specifying Port in the Login URL Field

While adding the identity provider, do not specify the login URL and clear the show card option. Use the login URL to access the service provider:

```
https://idp.sitea.novell.com/nidp/saml/idpsend?

PID = https://idp.siteb.novell.com/nidp/saml/metadata&

TARGET = https://idp.siteb.novell.com/nidp/app
```

In the identity provider, while adding the service provider, configure ID in the intersite transfer page. Configure the login URL with port number -2443 instead of the provider ID URL:

```
https://idp.sitea.novell.com:2443/nidp/saml/idpsend?
```

id = <idname>&TARGET=https://idp.siteb.novell.com:2443/nidp/app

Attributes Are Not Available Through Form Fill When OIOSAML Is Enabled

To workaround this issue, create a new attribute set with the OIOSAML mandatory attributes having remote attribute mapping as its OID equivalent and associate the attribute set to the identity provider configured at the SP.

Issue in Importing Metadata While Configuring Identity Provider or Service Provider Using Metadata URL

To work around this issue, manually copy the metadata by selecting the metadata text option while configuring an identity provider or service provider. The metadata text can be obtained from the browser.

Metadata Mentions Triple Des As Encryption Method

The OIO SAML metadata has tripledes-cbc and AES128-cbc mentioned as encryption methods. If triple des is not required, edit the following related tag in metadata and import the metadata manually.

```
Node:<md:EncryptionMethodAlgorithm="http://www.w3.org/2001/04/xmlenc#tripledes-cbc"/>
```

Issue in Accessing Protected Resources with External Identity Provider When Both Providers Use Same Cookie Domain

To workaround this issue, set 'agm.lagmode=false' in `/opt/novell/singlebox/mag/webapps/agm/agm.properties`.

SAML Intersite Transfer URL Setup Does Not Work for Non-brokered Setups after Enabling SP Brokering

To workaround this issue:

- Create a brokering group that has local IDP as Identity Provider and SP1 and SP2 as Trusted Providers.
- Create brokering rules for the Intersite Transfer URL requests to SP2. All requests to SP1 will be allowed.

Orphaned Identity Objects

When a persistent federation is configured or a transient federation with user mapping is configured by using Liberty, SAML 1 and SAML 2.0, the federation objects are created in the configuration store.When you delete or disable a user object, the objects in the configuration datastore related to this specific user become orphaned. These orphaned user profile objects affect the user lookup operations and system performances. These objects have to be removed manually using the Defed Tool: Federation Entry Management.

This tool clears all the orphaned federation objects related to Liberty, SAML 1, and SAML 2 from the trust and configuration datastore, except for Shared Secret entries.

NOTE: When the Access Manger setup includes Access Gateway and no persistent or transient federations have been configured, these objects are not created.

Linux:

1. Change the current working directory to `/opt/novell/devman/nam_tools/` from a terminal.
2. Run this command:

   ```
   /opt/novell/java/bin/java -classpath .:./lib/nam_tool.jar:./lib/nidp.jar:./lib/NAMCommon.jar:./lib/bcprov-jdk15-140.jar -Djava.util.logging.config.file=./conf/logging.properties com.novell.nam.tools.defed.DefedTool
   ```

3 The Defed tool will ask either to delete the orphan objects orexit from the tool. Select the option to delete the orphan objects. The tool will ask you to provide the IP address, port, user DN, and password.

4 The Defed tool deletes the orphaned federation objects and gives the summary of total number of federation entries encountered and number of the federation objects deleted.

This tool can be used to perform the operation on remote server too.

Windows:

1 Go to the `C:\Program Files (x86)\Novell\nam_tools` folder.

2 Run this command:

```
C:\Program Files (x86)\Novell\nam_tools>java -cp lib/nam_tool.jar;lib/
nidp.jar;lib/NAMCommon.jar;lib/bcprov-jdk15-140.jar -
Djava.util.logging.config.file=conf\logging.properties
com.novell.nam.tools.defed.DefedTool
```

3 Provide IP address, port, user DN, and password.

4 The Defed tool deletes the orphaned federation objects and gives the summary of total number of federation entries encountered and number of the federation objects deleted.

This tool can be used to perform the operation on remote server too.

Users cannot Log In to Identity Provider When They Access Protected Resources With Any Contract Assigned

To workaround this issue, ensure that the **Show Card** option is enabled on the default contract.

Attribute Query from OIOSAML.SP Java Service Provider Fails with Null Pointer

To workaround this issue:

1 Enable the OIOSAML compliance with service provider. The OIOSAML attribute set will be populated.

2 By default, the mandatory attributes are listed in the **Available** list.

3 Ensure that these mandatory attributes are moved from the **Available** list to the **Send with authentication** list to avoid the Null Pointer exception with OIOSAML compliance service providers.

Disabling the Certificate Revocation List Checking

For ADFS 2.0 to work with NetIQ Access Manger SAML 2.0, it is required to disable the Certificate Revocation List (CRL) checking.

To disable the CRL checking:

1 Modify the `tomcat7.conf` file of the Identity Server located at `/opt/novell/nam/idp/conf/tomcat7.conf`

2 Add this parameter`JAVA_OPTS="${JAVA_OPTS} -Dcom.novell.nidp.serverOCSPCRL=false"`

3 Restart the Identity Server by using this command: `/etc/init.d/novell-idp restart`.

Step up Authentication for the Identity Server Initiated SSO to External Provider Does not Work Unless It has a Matching Local Contract

For example, if a service provider is configured for a satisfiable contract that is only satisfiable by an external provider, then Intersite transfer service does not work.

To workaround this issue, ensure that any local contract is associated with the service provider. If not, then associate the same. External provider without any local contract is not supported

Metadata Cannot be Retrieved from the URL

To workaround this issue, verify the network card configuration for the proper DNS.

Requesting the Authentication to a Service Provider Fails

To workaround this issue:

1. In the Administration Console, click **Devices** > **Identity Servers** > **Edit** > **SAML 2.0** > **[Service Provider]** > **Authentication Response**.
2. Change **Artifact** to **Post** in the **Binding** field.

SAML 2.0 POST Compression Failure Does not Throw a Specific Error Code

The POST Compression feature is supported when both the identity provider and service provider understand SAML 2 POST deflate and inflate. If the service provider sends a compressed message, the identity provider needs to decompress the message and vice-versa. For the Access Manager identity server and service provider, the `nidpconfig.properties` file located in `/opt/novell/nam/idp/webapps/nidp/WEBINF/classes` needs to be modified to enable the SAML 2.0 POST deflate and inflate.

SAML 1.1 Service Provider Re-requests for Authentication

SAML 1.1 service provider performs a strict check on the name space of the attributes received in assertion.

To disable this, perform the following steps:

1. In the Administration Console, click **Devices** > **Identity Servers** > **Edit** > **Options** > **New**.
2. Select **SAML1X ATTRIBUTE MATCH BY NAME** in **Property Type**.
3. Select **true** in **Property Value**.
4. Click **OK** > **Apply**.

The Identity Server Statistics Logs Do Not Get Written In Less Than One Minute

The Identity Server statistics logs do not get written before one minute even though the time specified is less than one minute, for example, 10 seconds. This issue happens only when the time is specified to less than one minute.

Do not specify the time less than one minute. As a best practice, you should not set small period because it increases the load on the server and also increases the log file size exponentially.

No Error Message Is Written in the Log File When an Expired Certificate Is Used for the X509 Authentication

When a user tries to authenticate with an expired client certificate, the authentication fails. The log file does not have any information about the expiration of the certificate. Browsers also do not display any error message about it.

To see the logs related to expired certificates, perform the following steps:

1. Enable the following Java option in `tomcat7.conf` under `/opt/novell/nam/idp/conf/`:

   ```
   JAVA_OPTS="${JAVA_OPTS} -Djavax.net.debug=ssl,handshake"
   ```

   This option enables SSL logs.
2. Restart the Identity Server.

Terminating an Existing Authenticated User from the Identity Server

Access Manager provides the ability for users to single sign on to back end web servers. These back end web servers provide a series of protected resources that users can only access once authenticated to an Identity Server, and authorised by the Access Gateway. Having parsed the user credentials, and credentials validated against a back end user store, the Identity Server creates and maintains an active session for that user. Only when the user manually logs out of the Identity Server, or if the user's session timeout expires, then the user's active session will be removed. If the user continuously accesses protected resources before the session timeout expires, the session can remain active forever.

Use case: You may want to terminate an authenticated user sometimes. Some of the cases are listed below.

- User A who currently has an active session on the Identity Server and access to many protected resources, has had a designation change within the organization causing a change to resources that may be available. By forcing user A to logout and login again, his new roles or attributes may be retrieved by the Identity Server and used in policy evaluations by Access Manager to reflect his new position.
- User B who currently has an active session on the Identity Server and access to many protected resources, has been asked to leave an organization and all access to protected resources must be removed. By terminating user B's session on the Identity Server, any subsequent requests to the Identity Server will require the user to login again.

The **User Sessions** page in the Administration Console helps you to find users logged into your system and also helps to terminate their sessions if required. It displays the active user details for each Identity Server. You can search for a user with the user ID and terminate the sessions.

1. In the Administration Console Dashboard, click **Troubleshooting > User Sessions**.
2. Specify the user ID and click **Search**. If a match is found, it lists the IP address of the Identity Server and its sessions.
3. Click **Terminate Sessions** to terminate the sessions of the specific user.

   After you have performed the above procedure, the user sessions are terminated from the Identity Server, and any other trusted service providers it has provided an identity to during this session, for example, an Access Gateway or SAML 2.0 service provider.

NOTE: User details are fetched once per administration session. The last updated date is displayed. To refresh the data, click on **Refresh**.

X.509 Authentication Lists the Entire List of Certificates Imported to the Browser

To restrict the list to only certain certificates, use the following procedure:

1. Go to `etc/opt/novell/apache2/conf/cacerts/custom` and copy the required CA certificates manually to this folder using the following command:

   `cp <ca files in pem format> .`

   This command copies the CA certificates to the current folder.
2. Create a hash of the pem file using the following command:

   `openssl x509 -noout -hash -in <cafile.pem>`
3. Create a soft link in the same directory using the following command:

   `ln -s <cafile.pem> <hash value of the file>.0`

   For example, `ls -l` should display the following:

   ```
   /etc/opt/novell/apache2/conf/cacerts/custom # ls -ltotal 8lrwxrwxrwx 1 root root 22 2013-10-16 03:35 78038f2c.0 ->NAM-RP-Certificate.pem-rw-r--r-- 1 root root 5375 2013-10-16 03:31 NAM-RP-Certificate.pem
   ```
4. Restart Apache using the following command:

   `/etc/init.d/novell-apache2 restart`

Clustered Nodes Looping Due to JGroup Issues

In a cluster when multiple nodes are down, failover does not occur and user experiences looping due to jgroups issues.

Workaround: Modify the `/opt/novell/nids/lib/webapp/WEB-INF/web.xml` file on the Identity Server as follows to increase the jgroup timeouts.

```
<context-param> <param-name>JGroupsConfiguration</param-name> <param value>TCP(start_port=[nidp:ClusterPort];end_port=[nidp:ClusterPort][nidp:IfExternalAddress];external_addr=[nidp:ExternalAddress][nidp:EndIf]):TCPPING(initial_hosts=[nidp:ClusterMembers];port_range=1;timeout=3500;num_initial_members=2;up_thread=true;down_thread=true):MERGE2(min_interval=10000;max_interval=30000):FD_SOCK([nidp:IfExternalAddress]bind_addr=[nidp:ExternalAddress][nidp:EndIf]):FD(shun=true;timeou
```

```
t=5000;max_tries=5;up_thread=true;down_thread=true):VERIFY_SUSPECT(timeout=2000;do
wn_thread=false;up_thread=false):pbcast.NAKACK(down_thread=true;up_thread=true;gc_
lag=100;retransmit_timeout=3000):pbcast.STABLE(desired_avg_gossip=20000;down_threa
d=false;up_thread=false):pbcast.STATE_TRANSFER():pbcast.GMS(merge_timeout=10000;jo
in_timeout=5000;join_retry_timeout=2000;shun=true;print_local_addr=[nidp:DebugOn];
down_thread=true;up_thread=true)</param-value> </context-param>
```

For more information about the timeout options, see the following links:

- TCPPING
- MERGE2
- FD_SOCK
- FD
- VERIFY_SUSPECT
- NAKACK
- pbcast.STATE_TRANSFER
- pbcast.GMS

Authentication With Aliases Fails

If your userstore contains alias users and if you have configured alias class for authentication, the authentication fails. For the workaround, see TID 7015163.

nidp/app Does Not Redirect to nidp/portal after Authentication

After restarting the Identity Server, accessing `https://idp:port/nidp/app` first time does not redirect to `https://<idp-url>: port/nidp/portal`.

Workaround: After restarting the Identity Server, access User Portal first time by using `https://idp:port/nidp/`. Afterwards, the redirection works fine.

Login to Office 365 Fails when WS-Trust MEX Metadata Is Larger than 65 KB

To workaround this issue, perform the following steps:

1. Modify `/opt/novell/nam/idp/webapps/nidp/WEB-INF/web.xml` and map `mex.jsp` to the servlet mapping URL `/ndip/wstrust/sts/mex` as follows:

   ```
   <servlet> <servlet-name>NetIQSTS12MEX</servlet-name> <jsp-file>/jsp/mex.jsp</
   jsp-file> <load-on-startup>1</load-on-startup> </servlet><servlet-mapping>
   <servlet-name>NetIQSTS12MEX</servlet-name> <url-pattern>/wstrust/sts/mex</url-
   pattern> </servlet-mapping>
   ```

2. Restart the Identity Server.

Unsafe Server Certificate Change in SSL/TLS Renegotiations Is Not Allowed

After upgrading Access Manager from a version earlier than 4.0 Service Pack 1, if you have configured the Identity Server to point to the Load Balancer virtual IP address than the real IP addresses of the LDAP replica servers, the Identity Server's request to different LDAP server replicas fails.

The Identity Server health becomes yellow from green and displays the following warning:

```
Ensure that the following replicas are operating correctly XXXX
```

After validating the LDAP server replica, the following message is displayed:

```
Server certificate change is restricted during renegotiation
```

This happens because Access Manager uses JDK version 7u71 or later from the version 4.0 Service Pack 1 onwards. In JDK 7u71, unsafe server certificate change in SSL/TLS renegotiations is not allowed by default.

To workaround this issue, perform any one of the following actions:

- Add the following line in the `/opt/novell/nam/idp/conf/tomcat7.conf` file:

  JAVA_OPTS="${JAVA_OPTS} -Djdk.tls.allowUnsafeServerCertChange=true"
- Instead of specifying the load balancer virtual IP address as the LDAP replica server, ensure that the Identity Server refers to each LDAP server directly and not through the load balancer. In this way, the Identity Server maintains all communications with the LDAP servers directly, maintains states and connection information.
- Create a wildcard certificate and assign this server certificate to all the LDAP servers in the replica ring.

Viewing Request and Response Headers of All Protocols in a Log File

You can use one of the following options:

Option 1:

Perform the following steps:

1. Add the following filter to `/opt/novell/nam/idp/conf/web.xml`:

```
<filter>
<filter-name>requestdumper</filter-name>
<filter-class>
  org.apache.catalina.filters.RequestDumperFilter
</filter-class>
</filter>
<filter-mapping>
<filter-name>requestdumper</filter-name>
<url-pattern>*</url-pattern>
</filter-mapping>
```

2. Add the following to the `/opt/novell/nam/idp/conf/logging.properties` file:

```
# Dumper
1request-dumper.org.apache.juli.FileHandler.level = INFO
1request-dumper.org.apache.juli.FileHandler.directory = ${catalina.base}/logs
1request-dumper.org.apache.juli.FileHandler.prefix = request-dumper.
1request-dumper.org.apache.juli.FileHandler.formatter =
org.apache.juli.VerbatimFormatter
org.apache.catalina.filters.RequestDumperFilter.level = INFO
org.apache.catalina.filters.RequestDumperFilter.handlers = 1request-
dumper.org.apache.juli.FileHandler
```

3 Update the following handler in the `/opt/novell/nam/idp/conf/logging.properties` file:

```
handlers = 1catalina.org.apache.juli.FileHandler,
2localhost.org.apache.juli.FileHandler, 3manager.org.apache.juli.FileHandler,
4host-manager.org.apache.juli.FileHandler, java.util.logging.ConsoleHandler,
1request-dumper.org.apache.juli.FileHandler
```

A log file is created at `/var/opt/novell/nam/logs/idp/tomcat/` that contains a log of all the headers. A sample log file format: `request-dumper.2016-05-09.log`.

Option 2:

Add the following in `/opt/novell/nam/idp/conf/tomcat.conf`:

```
JAVA_OPTS="${JAVA_OPTS} -
Dcom.sun.xml.wss.provider.wsit.SecurityTubeFactory.dump=false"
JAVA_OPTS="${JAVA_OPTS} -Dcom.sun.xml.ws.transport.http.HttpAdapter.dump=true"
JAVA_OPTS="${JAVA_OPTS} -
Dcom.sun.xml.ws.transport.http.client.HttpTransportPipe.dump=true"
```

Troubleshooting Certificate Issues

- “Resolving Certificate Import Issues” on page 1000
- “Mutual SSL with X.509 Produces Untrusted Chain Messages” on page 1003
- “Certificate Command Failure” on page 1003
- “A Device Reports Certificate Errors” on page 1003
- “Renewing the expired eDirectory certificates” on page 1003

Resolving Certificate Import Issues

Use the following sections to resolve issues created when a full certificate chain is not imported in to Access Manager Appliance:

- “Importing an External Certificate Key Pair” on page 1001
- “Resolving a -1226 PKI Error” on page 1001
- “When the Full Certificate Chain Is Not Returned During an Automatic Import of the Trusted Root” on page 1002
- “Using Internet Explorer to Add a Trusted Root Chain” on page 1002

Importing an External Certificate Key Pair

The Access Manager Certificate Authority requires that all certificate key pairs in `.pfx` format contain the complete certificate chain. If a key pair was created with multiple CAs and the exported certificate does not contain the complete certificate chain, the file cannot be imported into Access Manager. When you try to import such a certificate, the following error message is displayed:

```
"Error importing certificate key pair: Error: Error: -1403
```

When exporting the certificate key pair, ensure that you include all the certificates in the certification path.

To ensure that your certificate contains all the intermediate certificates and contains them in the right order, import the certificate into Internet Explorer or Firefox.

- For Internet Explorer, click **Tools** > **Internet Options** > **Content** > **Certificates** > **Personal** > **Import**.
- For Firefox, click **Tools** > **Options** > **Advanced** > **Encryption** > **View Certificates** > **Your Certificates** > **Import**.

Make sure the browser contains the public key for all the intermediate CAs. Then select the certificate and export the certificate in `.pfx` format. In Internet Explorer, you must select to include all the certificates in the chain. In Firefox, all the certificates in the chain are automatically included.

If you receive an error when importing the certificate, the error comes from either NICI or PKI. For a description of these error codes, see Novell Certificate Server Error Codes and Novell International Cryptographic Infrastructure (http://www.novell.com/documentation/nwec/index.html).

Resolving a -1226 PKI Error

When you create a certificate signing request, send it to a third-party issuer to be signed, and receive the server certificate from the third-party issuer. You sometimes receive a -1226 error when you try to import the signed certificate. You receive this error when the issuer does not send the trusted roots required to validate the issuer of the server certificate.

Use one of the following options to resolve this issue:

- If the issuer included the trusted root and any intermediate certificates in a separate file or files, specify these files during the import by clicking the + character that allows you to add a trusted root or an intermediate certificate.
- If the issuer did not send you any additional files, you can go to the issuer's Web site, download them, then specify these files during the import by clicking the + character that allows you to add a trusted root or an intermediate certificate.
- You can try importing the certificate into Internet Explorer, which has the trusted roots from all major CAs, then export the certificate with the required chain of trusted roots. See "Using Internet Explorer to Add a Trusted Root Chain" on page 1002.

When the Full Certificate Chain Is Not Returned During an Automatic Import of the Trusted Root

Access Manager Appliance allows you to automatically import the trusted root under the following conditions:

- When enabling SSL communication between the Access Gateway and the Web server, you can automatically import the root CA from the Web server.
- When setting up the user stores for the Identity Server and adding the server replicas, you can automatically import the root CA of the LDAP server.

If there are multiple certificates in the chain, sometimes the server does not send all the certificates in the chain. When this happens, the following message is displayed:

```
The root CA certificate was not returned by the server. It might be necessary
to manually import the root CA certificate and possible intermediate CA
certificates in order to complete the chain.
```

To correct this problem, you need to manually import the missing entries. The easiest method to obtain all the certificates in the chain, including the root CA, is to import the server certificate into Internet Explorer, then export the chain and import it into Access Manager. If Access Manager already has some of the certificates, it skips their import and imports only the missing certificates.

For instructions on this process, see “Using Internet Explorer to Add a Trusted Root Chain” on page 1002.

Using Internet Explorer to Add a Trusted Root Chain

The following procedure works only when Internet Explorer contains the trusted root certificate of the issuer of your certificate.

1. In Internet Explorer, click **Tools** > **Internet Options** > **Content** > **Certificates**.
2. Click **Import** and import your server certificate into the **Other People** tab.
3. Click **Other People**, then double-click your certificate.
4. Click **Certification Path**.
   - If the **Certification Path** shows that the certificate is OK, you now have the full certificate chain available for export. Click **OK**, then continue with Step 5.
   - If the **Certification Path** is not OK, you cannot use this method. Click **OK**, then contact your issuer for the certificate chain.
5. Select the certificate, then click **Export** > **Next**.
6. Select **Cryptographic Message Syntax Standard - PKCS #7 Certificates (.P7B)** as the format and select **Include all certificates in the certification path if possible** to include the certificate chain.
7. Click **Next**, then specify a filename and path for the file.
8. Click **Next** > **Finish**.
9. Use this P7B file to import your server certificate into Access Manager.

Mutual SSL with X.509 Produces Untrusted Chain Messages

When you set up an X.509 contract for mutual SSL authentication, you must ensure that the Identity Server trust store (NIDP-truststore) contains the trusted root from each CA that has signed the client certificates. If a client has a certificate signed by a CA that is not in the Identity Server Trust Store, authentication fails.

To add a certificate to the Identity Server Trust Store:

1. In the Administration Console, click **Devices** > **Identity Servers** > **Edit** > **Security** > **NIDP Trust Store**.
2. Click either **Add** or **Auto-Import From Server** and follow the prompts.

Certificate Command Failure

Certificate commands are generated when you upgrade the Administration Console. Ensure that they have completed successfully.

1. To determine whether a certificate command has failed, click **Security** > **Command Status**.
2. Note the destination trust store or keystore of the failed command.
3. Click **Troubleshooting** > **Certificates**.

   The Certificates page displays all the keystores and trust stores configured for Access Manager.
4. Select the store, then click **Re-push certificates**.

   This sends all assigned certificates to Access Manager Appliance.

A Device Reports Certificate Errors

After you restore a device, especially the Administration Console, the device might report certificate errors. To fix these errors, you need to re-push the certificates from the Administration Console to the device:

1. Click **Troubleshooting** > **Certificates**.
2. Select the store that is reporting errors, then click **Re-push certificates**.

   You can select multiple stores at the same time.
3. (Optional) To verify that the re-push of the certificates was successful, click **Security** > **Command Status**.

Renewing the expired eDirectory certificates

The Secondary Administration Console stops working when the eDirectory certificates expire. When a certificate is about to expire, Access Manager shows a warning message when you log in to the Administration Console. You can check whether a certificate is expired on the Certificate Details page. See “Viewing Certificate Details” on page 807.

To workaround this issue manually renew the eDirectory certificates. For more information, see renewing the certificates.

Troubleshooting Access Manager Policies

This section discusses the following topics:

- “Turning on Logging for Policy Evaluation” on page 1004
- “Common Configuration Problems That Prevent a Policy from Being Applied as Expected” on page 1005
- “The Policy Is Using Old User Data” on page 1007
- “Form Fill and Identity Injection Silently Fail” on page 1009
- “Checking for Corrupted Policies” on page 1009
- “Policy Page Timeout” on page 1009
- “Policy Creation and Storage” on page 1009
- “Policy Distribution” on page 1010
- “Policy Evaluation: Access Gateway Devices” on page 1011

Turning on Logging for Policy Evaluation

Policy evaluation for roles occurs at the Identity Server. For Authorization and Identity Injection policies, policy evaluation occurs on the Embedded Service Provider (ESP) where the policy is enabled.

For the Form Fill policies, the evaluation and logging is done by ESP and the proxy service. To set the logging level on the Access Gateway for the proxy service, see “Enabling Form Fill Logging” on page 1041.

Logging for the policy evaluation done by ESP is controlled by the log settings of the Identity Server configuration. To enable this type of logging:

1. Click **Devices > Identity Servers** > *Edit* > **Logging**.

   If you have set up more than one Identity Server configuration, ensure that you select the configuration to which the other Access Manager Appliance components have been assigned.
2. Select **Enabled** for **File Logging**.
3. Select to echo the trace messages to the console: For the Access Gateway Appliance, Access Gateway Service, or Identity Server, this sends the messages to the `catalina.out` file.
4. (Optional) Specify a path for the Identity Server log files.
5. For policy evaluation tracing, set the **Application** level to **info** in the **Component File Logger Levels** section.

   If you are only troubleshooting policies at this time, do not select any other options. This reduces the amount of information recorded in the log files.

   To see the policy SOAP messages, you need to set the **Application** level to **verbose**.
6. Update the Identity Server.
7. Click **Auditing** > **General Logging** and download Identity Server and ESP catalina.out logs.
   - For role evaluation traces, view the Identity Server `catalina.out` file.

     If your Identity Servers are clustered, you need to look at the file from each Identity Server.
   - For Authorization, Form Fill, and Identity Injection evaluation traces, view the log file of ESP of the device that is protecting the resource.

Access Gateway Appliance or Service: This is the `catalina.out` file of the Access Gateway where the protected resource is defined. If the Access Gateway is part of a cluster, you need to look at this file from each Access Gateway in the group.

To view the actual ESP log file that contains only ESP log messages, see the `nidp.*.xml` files in the `/var/opt/novell/tomcat7/webapps/nesp/WEB-INF/logs` directory (or the directory you specified in step 4). Depending upon how you have configured **File Wrap**, the * portion of the filename contains the month, the week, the day, and the hour.

8 To understand what you are looking for in the log file, continue with one of the following:

- "Understanding Policy Evaluation Traces" on page 1028 if you set **Application** level to **info.**
- "Policy Evaluation: Access Gateway Devices" on page 1011 if you set **Application** level to **verbose**.

Common Configuration Problems That Prevent a Policy from Being Applied as Expected

When you try to determine what is functioning incorrectly in a policy, you need to turn on policy tracing and understand the evaluation traces. See the following:

- "Turning on Logging for Policy Evaluation" on page 871
- "Understanding Policy Evaluation Traces" on page 1028

The CO entry line of a policy trace identifies when a policy condition evaluates to False or True. The PA entry line indicates whether the Action was applied or ignored. If the results of the policy trace are not what you expected for the user, the next step is to determine why the policy isn't behaving the way you want it to. Check for the following problems:

- "Enabling Roles for Authorization Policies" on page 1005
- "LDAP Attribute Condition" on page 1006
- "Result on Condition Error Value" on page 1006
- "An External Secret Store and Form Fill" on page 1007

Enabling Roles for Authorization Policies

If you are using roles in your authorization policies, you need to ensure that the role is enabled for the Identity Server configuration. You can create roles and authorization policies independently of assigning them to protect a resource or to an Identity Server configuration.

If you have not enabled the role, users are not assigned the role when they log in, even when they meet all the criteria for the role.

- If the Authorization Policy is an Allow policy, the users might be denied access because they haven't been assigned the role.
- If the Authorization Policy is a Deny policy, the users might be allowed access because they haven't been assigned the role.

Whenever an Authorization Policy is not producing the expected results and the policy contains a role, the first troubleshooting step should always be to check whether the role has been enabled for the Identity Server configuration. Click **Devices > Identity Servers** > *Edit* > **Roles**. If the role is not enabled, the Identity Server cannot assign the role to the user.

The second step should be to ensure that the roles are transferred from for Identity Server to the Embedded Service Provider. Click **Devices > Identity Servers** > *Edit* > **Liberty > Web Service Provider**. The **Authentication Profile** needs to be enabled in order for Embedded Service Providers to evaluate roles in policies. This profile is enabled by default, but it can be disabled. When it is disabled, all devices assigned to use this Identity Server cluster configuration cannot determine which roles a user has been assigned, and the devices evaluate policies as if the user has no roles.

LDAP Attribute Condition

If you use an LDAP attribute as the condition for a Role policy or an Authorization policy and your users are not being assigned the role or are denied access to a resource, the most likely cause of the problem is the LDAP attribute name used in the policy. Some administration tools for the LDAP user stores display a UI name or an eDirectory™ name rather than the LDAP attribute name. Access Manager Appliance policies require the LDAP attribute name.

Use the following steps to identity whether the Access Manager Appliance policy has been configured for the LDAP attribute name, a UI name, or an eDirectory name:

1. Use an LDAP browser to view one of your users in your LDAP user store.

   You can download a Java-based tool from the Internet.
2. Verify the LDAP name of the attribute and that the user has the expected value.
3. In the Administration Console, click **Policies** > **Policies** > **[Name of Policy]** > **Rule Number**.
4. View the attribute name and value for the LDAP Attribute condition.
5. Verify the following:
   - The name of the attribute should match the name as displayed in the LDAP browser. The attribute name is not case sensitive, but it should not contain any spaces. If you need to modify the attribute used by the policy, click the attribute name, then select an attribute from the list or select **New LDAP Attribute** to add one.
   - The value can be case sensitive, depending upon how you have configured the **Mode** for the policy. If you have selected case sensitive for the **Mode**, ensure that the case in the policy matches the case in the LDAP user store.
   - If the attribute is multi-valued and your users typically have multiple values, select **Substring** as the **Comparison** type.
6. If these steps have not solved the problem, see “Result on Condition Error Value” on page 1006.

Result on Condition Error Value

If you incorrectly set the value of the **Result on Condition Error** field, you create a policy that allows an action that you want the policy to deny or that denies an action that you want allowed. You must carefully evaluate whether you want the action applied or ignored when an error occurs during the evaluation of the condition. For positive conditions, the following rules apply:

- For the action to be applied, either the user must match the condition or the **Result on Condition Error** must be set to True.
- For the action to be ignored, either the user must not match the condition or the **Result on Condition Error** must be set to False.

The logic is harder to follow when you start adding “if not” to the conditions. The user then matches the condition by not matching the condition. For this type of condition, you need to ask whether you want the action applied to any user when an error occurs evaluating the condition.

The logic is even harder to following when you start adding multiple condition groups that can also have “or nots” and “if nots”.

If you have a policy that uses “if not” conditions or uses multiple condition groups and it is not producing the expected results, you might want to rewrite the policy so that it contains only positive conditions.

You might want to modify the condition groups so that the policy uses multiple rules, with each rule containing one condition group with the conditions you want the user to match for the action you assign to the rule.

An External Secret Store and Form Fill

When you create a user store on the Identity Server (**Local > User Stores**) and define it as an external Secret Store (**Liberty > Web Service Provider > Credential Profile**), some attributes are not being created properly on the SAML affiliate object. The workaround is to access the user store configuration page (**Local > User Stores**), then exit. This action results in a check to verify that the schema, objects, and attributes exist, and the affiliate object is then re-created from scratch, if necessary.

The following affiliate objects must exist:

```
authsamlCertContainerDN (container holding trusted certificates,
   for example: SCC Trusted Root.Security)
authsamlProviderID
authsamlTrustedCertDN (list of trusted certificate(s))
authsamlValidAfter (180 seconds default)
authsamlValidBefore (180 seconds default)
```

If these attributes exist, the system works normally. However, if your Identity Server and Secret Store server are not configured to use the same NTP server, time synchronization can be a problem. If time synchronization is an issue, you can change the 180-second default validity times as a workaround.

If your LDAP user store and the Administration Console have a firewall separating them, TCP ports 524 and 636 must be open to allow for the creation of the required objects. For more information about ports and firewalls, see Setting Up Firewalls in the NetIQ Access Manager Appliance 4.2 Installation and Upgrade Guide.

The Policy Is Using Old User Data

When a policy is first evaluated, it caches information about the user.

- Some data items are updated every minute.
- Some are cached for the duration of the request.
- Some are cached for the duration of the user’s session. When a data item is cached for the duration of a user session, the user must log out and log in for the policy modification to take effect.

Table 26-2 lists how long the data items for a condition are cached before being refreshed.

Table 26-2 *Data Caching Limits*

Condition	Data Refresh Interval
Authenticating IDP	User session

Condition	Data Refresh Interval
Authentication Contract	User session
Authentication Method	User session
Authentication Type	User session
Client IP	Request
Credential Profile	User session
Current Date	One minute
Current Day of Week	One minute
Current Day of Month	One minute
Current Time of Day	One minute
HTTP Request Method	Request
Java Data Injection Module	User session
LDAP Attribute	User session; configurable to be cached only for the request with the Force Data Read option.
LDAP Group	User session
LDAP OU	User session
Liberty User Profile	User session
Proxy Session Cookie	User session
Roles for Current User	User session
Roles from Identity Provider	User session
Shared Secret	User session; configurable to be cached only for the request with the Force Data Read option.
String Constant	User session
URL	Request
URL Scheme	Request
URL Host	Request
URL Path	Request
URL File Name	Request
URL File Extension	Request
User Store	User session
X-Forwarded-For IP	Request

Form Fill and Identity Injection Silently Fail

Login with Form Fill or Identity Injection can fail when all of the following conditions occur:

- Your user store is configured to use Novell® SecretStore®.
- The shared secrets needed for Form Fill or Identity Injection are locked because the shared secrets are used by another application that is using the enhanced security feature. For example, if the application writes a secret called ssn, and you use that same secret in a Form Fill or Identity Injection policy, that secret is locked whenever the admin changes the user's password. Access Manager Appliance does not use the enhanced security feature when it writes shared secrets.

The new unlock feature for SecretStore can resolve this issue. See "Determining a Strategy for Unlocking the SecretStore" on page 280.

Checking for Corrupted Policies

For a policy to be evaluated correctly, the policy must contain a rule. To verify that your system does not contain any policies with configuration errors:

1. In the Administration Console Dashboard, click **Troubleshooting** > **Policies**.

   If you have any corrupted policies, they appear in the list.
2. Identify the corrupted policy, then click **Remove**.

Policy Page Timeout

If your policy page hangs, and you have an LDAP group or LDAP ou being used in the policy, check the health of your user stores (LDAP servers) and ensure that they are communicating.

Policy Creation and Storage

For troubleshooting, you can export the policy and send it to NetIQ for debugging. If the policy uses roles, ensure that you also export the Role policies.

Policies are stored as XML documents in the object directory, with one XML document to represent each policy container. The default policy container (Master_Container) resides at:

```
\\novell\accessManagerContainer\VCDN_Root\PartitionsContainer\Partition\ContentPublisherContainer\mastercdn\xpemlPEP\romaContentCollectionXMLDoc
```

Other policy containers are stored following the same path, with a unique name string representing the policy name that replaces the ou=mastercdn portion of the above path.

If you are unsure if the policy is being created correctly or if you need to check to see if the policy is enabled, you can view the policy list in the interface. If you think the GUI is not properly displaying the policy, you can also view the XML by navigating to the Policy Conditions on which you edit rules, right click and choose **This Frame > View Frame Source**.

Policy Distribution

Policy definitions are not replicated, but are referenced by the Access Gateways for which the policy is to be evaluated. The policy reference mechanism is a set of XML elements that refer back to the policy definitions stored in the various policy containers. If you have configured a policy for a protected resource and an Access Gateway does not seem to be executing this policy, use the following procedures to verify that the Access Gateway has been configured to use the policy:

1. Set the level of Application logging to **verbose**. See "Turning on Logging for Policy Evaluation" on page 871.

   This enables the tracing of the policy enforcement lists.

2. Search for name of your policy in a `<PolicyEnforcementList>` element. The ExternalElementRef attribute contains a reference to the policy name.

   You can find these elements in the `catalina.out` file.

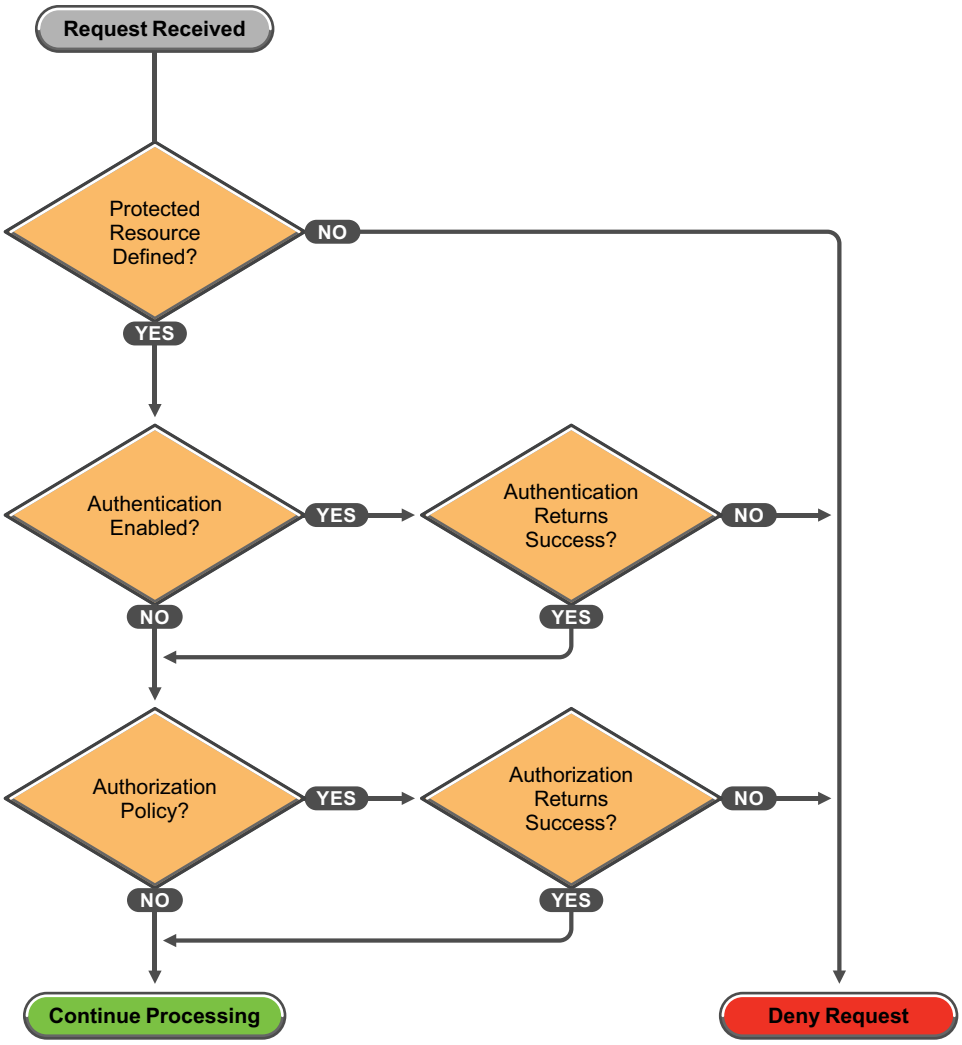
3. If you cannot find the policy name, the Access Gateway has not been configured to use the policy. The configuration either needs to be applied or the policy needs to be enabled. For information about how to assign a policy to a protected resource, see "Configuring Protected Resources" on page 101.

4. If you find the policy name associated with the correct protected resource, you need to check why the policy is not evaluating according to your design. Set the level of Application logging to **info** and examine the policy trace from a user accessing the protected resource. See "Understanding Policy Evaluation Traces" on page 1028.

Policy Evaluation: Access Gateway Devices

The following diagram depicts how Authorization policies fit into the protected resource processing for the proxy.

Figure 26-8 *Policy Evaluation*



The SOAP messages are output to the `catalina.out` file. Sample SOAP messages are shown in the following scenarios:

- “Successful Policy Configuration Example” on page 1011
- “No Policy Defined Configuration Example” on page 1012
- “Deny Access Configuration/Evaluation Example” on page 1013

Successful Policy Configuration Example

Note the Policy Enforcement Point (PEP) identifier of AGIdentityInjection in the request and the PolicyID in the response.

Configuration Request

```
toBufSeg: <?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/
   envelope/">
<SOAP-ENV:Body>
   <NXPES ID="12">
      <Configure-ag PEPName="AGIdentityInjection">
         <PolicyEnforcementList
            RuleCombiningAlgorithm="DenyOverridesWithPriority"
            schemaVersion="1.32"
            LastModified="1138389868885"
            LastModifiedBy="cn=admin,o=novell">
            <PolicyRef ElementRefType="ExternalWithIDRef"
                ExternalElementRef="PolicyID_xpemlPEP_AGIdentity
                    Injection_ii_test"
                ExternalDocRef="ou=xpemlPEP,ou=mastercdn,
                    ou=ContentPublisherContainer,ou=Partition,
                    ou=PartitionsContainer,ou=VCDN_Root,ou=access
                    ManagerContainer,o=novell:romaContentCollection
                    XMLDoc"
                UserInterfaceID="PolicyID_xpemlPEP_AGIdentity
                    Injection_ii_test"/>
         </PolicyEnforcementList>
      </Configure-ag>
   </NXPES>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Configuration Response

```
LibertyProcessMsgCB:
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
<SOAP-ENV:Body>
   <NXPES Id="" Status="success">
      <ConfigureResponse PolicyId="755OK8P0-7543-518M-8L8M-N0P2LM2
               N3O27">
         <ContextDataElement Enum="2551"/>
      </ConfigureResponse>
   </NXPES>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

No Policy Defined Configuration Example

The following is a sample of a configuration request where the policy code detects that no policies are in effect for the protected resource and Policy Enforcement Point (PEP).

Configuration Request

```
toBufSeg: <?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
<SOAP-ENV:Body>
   <NXPES ID="11">
      <Configure-ag PEPName="AGAuthorization">
         <PolicyEnforcementList
            RuleCombiningAlgorithm="DenyOverridesWithPriority"
            schemaVersion="1.32"
            LastModified="1138389868885"
            LastModifiedBy="cn=admin,o=novell">
           <PolicyRef ElementRefType="ExternalWithIDRef"
               ExternalElementRef="PolicyID_xpemlPEP_AGIdentity
                        Injection_ii_test"
               ExternalDocRef="ou=xpemlPEP,ou=mastercdn,ou=Content
                       PublisherContainer,ou=Partition,ou=Partitions
                       Container,ou=VCDN_Root,ou=accessManager
                       Container,o=novell:romaContentCollectionXMLDoc"
               UserInterfaceID="PolicyID_xpemlPEP_AGIdentityInjection_
                       ii_test"/>
         </PolicyEnforcementList>
      </Configure-ag>
   </NXPES>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Configuration Response

```
LibertyProcessMsgCB:
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/
      envelope/">
   <SOAP-ENV:Body>
      <NXPES Id="" Status="emptypolicyset"/>
   </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Deny Access Configuration/Evaluation Example

The following is a sample of a configuration request for a Deny policy and an evaluation request for this policy.

Configuration Request

```
toBufSeg: <?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/
   envelope/">
<SOAP-ENV:Body>
   <NXPES ID="17">
      <Configure-ag PEPName="AGAuthorization">
         <PolicyEnforcementList
            RuleCombiningAlgorithm="DenyOverridesWithPriority"
            schemaVersion="1.32"
            LastModified="1138718667305"
            LastModifiedBy="cn=admin,o=novell">
         <PolicyRef
            ElementRefType="ExternalWithIDRef"
            ExternalElementRef="PolicyID_xpemlPEP_AGIdentityInjection
                _custom_test"
```

```
            ExternalDocRef="ou=xpemlPEP,ou=mastercdn,ou=Content
               PublisherContainer,ou=Partition,ou=PartitionsContainer,
               ou=VCDN_Root,ou=accessManagerContainer,o=novell:roma
               ContentCollectionXMLDoc"
            UserInterfaceID="PolicyID_xpemlPEP_AGIdentityInjection
               _custom_test"/>
         <PolicyRef
            ElementRefType="ExternalWithIDRef"
            ExternalElementRef="PolicyID_xpemlPEP_AGAuthorization_
               deny-all"
            ExternalDocRef="ou=xpemlPEP,ou=mastercdn,ou=Content
               PublisherContainer,ou=Partition,ou=PartitionsContainer,
               ou=VCDN_Root,ou=accessManagerContainer,o=novell:roma
               ContentCollectionXMLDoc"
            UserInterfaceID="PolicyID_xpemlPEP_AGAuthorization
               _deny-all"/>
         </PolicyEnforcementList>
      </Configure-ag>
   </NXPES>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Configuration Response

```
LibertyProcessMsgCB:
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/
    envelope/">
<SOAP-ENV:Body>
   <NXPES Id="" Status="success">
      <ConfigureResponse
           PolicyId="55N3NL81-L29N-2619-K0M8-2L963M0MM701"/>
   </NXPES>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Evaluation Request

```
toBufSeg: <?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
<SOAP-ENV:Body>
   <NXPES ID="18">
      <Evaluate PolicyId="55N3NL81-L29N-2619-K0M8-2L963M0MM701"
                Verbose="on"/>
   </NXPES>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Evaluation Response

```
LibertyProcessMsgCB:
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/
      envelope/">
<SOAP-ENV:Body>
   <NXPES Id="" Status="success">
      <EvaluateResponse>
         <DoAction ActionName="Deny" ActionTTL="-1" Enum="2620">
            <Parameter Enum="10" Name="Message" Value=""/>
         </DoAction>
      </EvaluateResponse>
   </NXPES>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Troubleshooting MobileAccess

This section discusses how to troubleshoot issues that occur when using MobileAccess.

Using the Same Mobile Device for Different Users Causes the Expired Session Error

Issue: You have a company iPad or Android tablet. User1 has registered and used the tablet with the company and then left the company. You deregister User1 and then reassign the tablet to User2. After User 2 registers the tablet and tries to access an appmark, they get an error of expired session.

Solution: The user must try the appmark a second time and then they can access the resource. The reason for this is a cookie still exists for User1, who no longer is valid. The second attempt replaces the cookie with a valid cookie for User2.

Simple Authentication with a Pop-up Browser Window Does Not Work for MobileAccess

Issue: You have a web server configured to use simple authentication. Simple authentication requires a pop-up browser window for authentication. The users receives a 401 error.

Solution: Mobile platforms do not support pop-up browser windows. Any web server that requires a simple authentication, using a pop-up browser window, fails in the mobile environment with the internal browser. DO not use pop-up browsers windows for authentication in a MobileAccess environment.(948904)

Users Fail to Authenticate to MobileAccess when Appmarks Are Launched in the Chrome Browser

Issue: If a user sets up a Chrome profile and then tries to use a Google Apps resource configured to use Chrome on MobileAccess, the login fails because Chrome passes the saved profile user name and password to the resource instead of passing the user name and password from MobileAccess. This issue occurs for any Google Apps resource (for example, Gmail or Google Drive) on iOS and Android mobile devices. (Bug 948622)

Solution: Users can remove their Chrome profile to avoid this issue, or you can configure the appropriate Google Apps appmarks in CloudAccess so the resources open with Firefox, an internal viewer, or a user-selectable option, instead of Chrome.

Troubleshooting Code Promotion

- "Troubleshooting Identity Server Code Promotion" on page 1016
- "Troubleshooting Access Gateway Code Promotion" on page 1016
- "Troubleshooting Device Customization Code Promotion" on page 1019

Troubleshooting Identity Server Code Promotion

This section discusses how to troubleshoot any issue occurred during Identity Server Code Promotion.

Importing Identity Server Configuration Data Fails

Error message: `Configuration Import Failed`

While importing the configuration data, the Import Configuration wizard displays this message.

See the details of the failure the Administration Console tomcat logs at the following location:

`/opt/novell/nam/adminconsole/logs/catalina.out`

Collect the error details and contact the Technical Support team.

To restore your system, go to the **Dashboard** and click the drop-down menu in the upper right corner > **Code Promotion**. You will find the backup file that was created as part of import. Download the file and then click **Import Configuration** on the same page. Re-import this backup configuration to restore to the previous configuration.

Troubleshooting Access Gateway Code Promotion

This section discusses how to troubleshoot any issue occurred during Access Gateway Code Promotion.

- "Importing Access Gateway Configuration Data Fails" on page 1017
- "Policy Configuration Is Locked" on page 1017
- "Access Gateway Configuration Is Locked" on page 1017
- "Access Gateway Cluster Is Not Associated with any Identity Server" on page 1017
- "Proxy Service Type Does Not Match" on page 1018
- "Policy Type Does Not Match" on page 1018
- "Cannot Import a Virtual Proxy Service to SSL enabled Master Proxy" on page 1018
- "Cookie Domain and Published DNS Name Do Not Match" on page 1018
- "SSL Enabled Web Server Configuration Is Imported to a Non-SSL Proxy Service" on page 1018
- "Names of Master Proxy Service Are Different" on page 1019
- "Reverse Proxy and Master Proxy Service Do Not Exist" on page 1019
- "Proxy Service Does Not Exist in the Target Setup" on page 1019

- “DNS Name Is Not Unique” on page 1019
- “Revert Process Fails for Access Gateway” on page 1019

Importing Access Gateway Configuration Data Fails

Error message: `Configuration Import Failed`

While importing the configuration data, the Import Configuration wizard displays this message.

See the details of the failure the Administration Console tomcat logs at the following location:

`/opt/novell/nam/adminconsole/logs/catalina.out`

Collect the error details and contact the Technical Support team.

You can restore the Access Gateway configuration by using the backup file if you have backed up the configuration by using the `ambackup` file.

Policy Configuration Is Locked

Error message: `Policy configuration locked by another user`

If an administrator is making changes to policies and you try to import the configuration by using Code Promotion simultaneously, then import fails.

Ensure that while importing, no other administrator is making changes to configuration. If it is already locked, click **Please unlock to override**.

You also need to check which policy containers are locked and then unlock them from the Policy user interface.

Access Gateway Configuration Is Locked

Error message: `Access Gateway configuration locked by another user`

If an administrator is making changes to Access Gateway configuration and you try to import the configuration by using Code Promotion simultaneously, then import fails.

Ensure that while importing, no other administrator is making changes to configuration. If it is already locked, click **Please unlock to override**. Unlock the Access Gateway cluster in the Access Gateway user interface for which you are importing the configuration data.

Access Gateway Cluster Is Not Associated with any Identity Server

Error message: `Could not generate Access Gateway import overview`

Ensure that you associate the Access Gateway cluster with an Identity Server cluster before importing protected resources that have Identity Server dependencies such as contracts and custom attributes.

Proxy Service Type Does Not Match

Error message: `Proxy service name not unique`

If the name of a proxy service is same on the source and target systems, but their type does not match, then the import does not happen. For example, a proxy service is Path Based Multi-Homing on the source setup and a proxy service with the same name is Domain Based Multi-Homing on the target system.

Update the type of the proxy service on the source setup or target setup and then import.

Policy Type Does Not Match

Error message: `Invalid input`

`Type Mismatch Error: Cannot import policy <name of the policy> of container <name of the container>. The type of this policy is <type of policy> in the source setup and <type of policy> in the target setup.`

If the name of a policy is same on the source and target systems, but their type does not match, then the import does not happen. For example, a policy is defined as authorization policy in the source setup and a policy with the same name is defined as identity injection in the target setup.

Update the type of the policy on the source setup or target setup and then import.

Cannot Import a Virtual Proxy Service to SSL enabled Master Proxy

Error message: `Invalid input`

`Cannot import the new virtual proxy service in (name of reverse proxy) > (name of proxy service) from source Access Gateway cluster <name of the cluster> because SSL is enabled in the reverse proxy <name of the reverse proxy on the target system> in the target Access Gateway cluster.`

Import of virtual proxy services to a SSL enabled proxy service in the target system is not allowed.In such cases, ensure that you exclude virtual proxy services during import.

Cookie Domain and Published DNS Name Do Not Match

Error message: `Domain-Based Multi-Homing requires the Published Domain Name of proxy service <name of the proxy service being imported> to be in the Cookie Domain of the first Proxy Service under Reverse Proxy`

Master proxy service's cookie domain does not match with the imported Domain Based Proxy Service's DNS name.

Update the published DNS name for the specified proxy service while importing it.

SSL Enabled Web Server Configuration Is Imported to a Non-SSL Proxy Service

Error message: `Invalid input`

`Cannot import the SSL enable proxy service in (name of reverse proxy) > (name of proxy service) from the source Access Gateway cluster because SSL is not enabled in the reverse proxy in the target Access Gateway cluster`

You cannot import SSL enabled proxy service to non SSL enabled reverse proxy. Before importing, enable SSL for the target reverse proxy or disable SSL for source proxy service.

Names of Master Proxy Service Are Different

Error message: `Invalid input`

```
Cannot import master proxy service from the source Access Gateway cluster as
another master proxy service with a different name already exists in the target
Access Gateway cluster.
```

Name of the master proxy service must be same on the source and target systems. Update the name on the source or target setup before importing it.

Reverse Proxy and Master Proxy Service Do Not Exist

Error message: `Invalid input`

```
Reverse Proxy does not exist in the target Access Gateway cluster
```

For importing a proxy service or protected resource, if the corresponding reverse proxy or master proxy service does not exist, then you must create reverse proxy and master proxy service on the target system before starting the Code Promotion import.

Proxy Service Does Not Exist in the Target Setup

Error message: `Invalid input`

```
Proxy Service does not exist in the target Access Gateway cluster
```

Importing only selected protected resources for a domain-based proxy service that does not exist in the target setup fails. You must also import the related domain-based proxy service in such cases.

DNS Name Is Not Unique

Error message: `Published DNS Name is not unique under Reverse Proxy in the target setup`.

DNS name must be unique under a reverse proxy. Specify a unique name in the **Published DNS Name** field for the proxy service during import.

Revert Process Fails for Access Gateway

In case of any error during the import process, system tries to revert to the previous configuration. If any error occurs during this revert process, then Code Promotion displays a message specifying the component for which the revert process failed. Components include Access Gateway configuration and dependent policies and policy extensions. In this case, you need to restore the pre-import configuration manually by using `ambackup`. You should take a backup by using the `ambackup` file before importing the configuration data.

Troubleshooting Device Customization Code Promotion

This section discusses how to troubleshoot any issue occurred during device customization Code Promotion.

Custom Files Are Not Imported

Ensure that the custom files are available in the source setup and paths are correct.

Verify Administration Console `catalina.log` of the source setup after export. This log file contains information about files which are not exported.

Troubleshooting OAuth and OpenID Connect

This section discusses the following issues and workaround:

- "Users Cannot Register a Client Application" on page 1020
- "Token Exchanges Show Redirect URI Invalid Error" on page 1020
- "Users Cannot Register or Modify a Client Application with Specific Options" on page 1020
- "A Specific Claim Does Not Come to the UserInfo Endpoint during Claims Request" on page 1021
- "Access Gateway OAuth Fails" on page 1021
- "After Allowing Consent, 500 Internal Server Error Occurs" on page 1021
- "No Error Message When a Token Request Contains Repetitive Parameters" on page 1021
- "Oauth Token Encryption/Signing Key Is Compromised or Corrupted" on page 1021
- "Tracing OAuth Requests" on page 1022
- "OAuth Client Registration Fails If a Role Policy Contains a Condition Other than LDAP Attribute, LDAP Group, or LDAP OU" on page 1022
- "The Identity Injection Policy Does Not Inject Passwords" on page 1023
- "OAuth Endpoint Generates Access and Refresh Tokens Using OAuth Resource Owner Credential Flow" on page 1023
- "OAuth Apps Fail After Upgrading Access Manager" on page 1023

Users Cannot Register a Client Application

In the Administration Console, verify whether the user has role `NAM_OAUTH2_DEVELOPER` or `NAM_OAUTH2_ADMIN` configured in the Identity Server Role policy configuration.

Verify the REST communication between browser and Identity Server by using Chrome developer console.

Token Exchanges Show Redirect URI Invalid Error

Go to **Administration Console** > **Devices** > **Identity Servers** > **Edit** > **OAuth & OpenID Connect** > **Client Applications**. Open the client application and verify whether the specified URI is configured for the client application.

Users Cannot Register or Modify a Client Application with Specific Options

Verify the options enabled for the client application. An administrator must enable same options in the Global Settings page.

A Specific Claim Does Not Come to the UserInfo Endpoint during Claims Request

Verify the following points:

- Whether the user has a value for that attribute. If the value is empty, UserInfo does not return any value in JSON.
- The Identity Sever has provided the requested scope. You can check this with the TokenInfo endpoint by providing an Access token.
- The scope contains the attribute mapping for the missing attribute.

  You can access only LDAP attributes as claims at this time.

Access Gateway OAuth Fails

Perform the following actions:

- Verify whether **Activate OAuth** is selected for the Protected Resource.
- Verify authorization policies are configured. Also, verify if the token contains required scopes by using the TokenInfo endpoint.
- Verify Identity Injection policies. Enable Application debug logs in the Identity Server and ESP and check for policy results.

After Allowing Consent, 500 Internal Server Error Occurs

Verify whether the attribute you have configured in the Global Setting page is available and stored in the user store. Ensure that the correct attribute to store authorization grant is available in the user store and it is writable to the user store.

No Error Message When a Token Request Contains Repetitive Parameters

Ensure that you do not send the same parameter multiple times in a single request. The base framework reads only last or first available parameters if multiple query parameters have the same name.

Oauth Token Encryption/Signing Key Is Compromised or Corrupted

Regenerate the token encryption/signing key by using the following steps:

1. Delete the `nidsOAuthKeysXML` attribute from the e-Directory at the following location:

   `o=novell, ou=accessManagerContainer, cn=nids, cn=cluster, cn=IDP_Cluster, cn=OAuth_Container`
2. Go to the Administration Console and click **Devices** > **Identity Server** > **Edit**.
3. Update the Identity Server cluster.

Tracing OAuth Requests

You can trace each OAuth request and response by setting the following property in `tomcat.conf`.

Add the following line in `/opt/novell/nam/idp/conf/tomcat.conf`:

```
JAVA_OPTS="${JAVA_OPTS} -Dcom.novell.nidp.oauth.jersey.trace=ALL"
```

You can specify the following parameters:

- OFF: tracing support is disabled (default value)
- ON_DEMAND: tracing support in a stand-by mode. It is enabled selectively per request through a special X-Jersey-Tracing-Accept HTTP request header. The Jersey tracing facility does not use the value of the X-Jersey-Tracing-Accept header and as such, it can be any arbitrary string.
- ALL: tracing support is enabled for all requests

You can customize the level of detail of the information (tracing threshold) provided by Jersey tracing facility. You can set the tracing threshold at a request level through X-Jersey-Tracing-Threshold HTTP request header. The request level configuration overrides any application level setting. Supported levels include SUMMARY, TRACE, and VERBOSE.

- **SUMMARY**: basic summary information about the main request processing stages
- **TRACE**: detailed information about activities in all main request processing stages (default threshold value)
- **VERBOSE**: extended information similar to the TRACE level, however it includes details about entity providers (MBR/MBW) that were skipped during the provider selection phase for any reason (such as lower priority or pattern matching). Additionally, in this mode all received request headers are echoed as part of the tracing information.

For more information, see Monitoring and Diagnostics.

OAuth Client Registration Fails If a Role Policy Contains a Condition Other than LDAP Attribute, LDAP Group, or LDAP OU

For registering OAuth client applications by using the Identity Server, you must have a role called `NAM_OAUTH2_DEVELOPER` assigned.

The following are the recommended conditions in an Identity Server Role policy that assigns the `NAM_OAUTH2_DEVELOPER` role:

- LDAP Attribute
- LDAP Group
- LDAP OU conditions

The client registration will not work if this role policy contains any of the following conditions:

- Authenticating IDP
- Authentication Contract
- Authentication Method
- Authentication Type
- Credential Profile

- Liberty User profile
- Roles from Identity Provider
- User Store

The Identity Injection Policy Does Not Inject Passwords

Verify logs by enabling the debug level. Verify whether, in the Identity Server, the userinfo request is coming with sp-id. Logs should include the `fetching password for user` term. If any issue occurs, the log includes the error message.

OAuth Endpoint Generates Access and Refresh Tokens Using OAuth Resource Owner Credential Flow

The OAuth token endpoint generates access and refresh tokens using the OAuth resource owner credential flow when you specify a valid user name and a null password.

To fix this issue, check the eDirectory configuration. The configuration must not allow any anonymous logins. In eDirectory, you must validate **ldapBindRestrictions: 0 (No Restriction)**

OAuth Apps Fail After Upgrading Access Manager

The OAuth apps fail after you upgrade Access Manager. This is caused due to the expired authorization code.

To workaround this issue, you need to upgrade both Access Gateway and Identity Server to Access Manager at the same time. For more information, see TID.

Troubleshooting User Attribute Retrieval and Transformation

This section discusses how to troubleshoot any issue occurred in User Attribute Retrieval and Transformation feature.

In Identity Provider No Value is Fetched from Attribute Source

In the Identity provider, no Attribute Source value is fetched from the data source. However in the Administrative Console, the test functionality fetches the data and displays the correct result.

To troubleshoot this issue, check the error logs for any Data Source connection issues.

- Linux: `/opt/novell/nam/logs/idp/tomcat/catalina.out`
- Windows: `\Program Files (x86)\Novell\Tomcat\logs\stdout.log`

If the Data Source is a database, make sure the driver jars are available in both Identity provider and Administration Console. If the Data Source is a secure LDAP connection, make sure SSL certificate of the LDAP directory is imported in the Identity provider's trust store. Update all Identity Servers.

Error Message While Testing a Database Connection

Error 1:

While testing a database connection, if the error message displays: `"Driver com.microsoft.sqlserver.jdbc.SQLServerDriver claims to not accept JDBC URL....."`, then the issue can be due to the format of the specified URL.

To troubleshoot this issue, specify a correct format of the JDBC URL. You can refer to the example specified in the Edit Data Source interface.

Error 2:

While testing a database connection, if the error message displays: "`driverClassName specified class 'com.microsoft.sqlserver.jdbc.SQLServerDriver' could not be loaded`", or "`driverClassName specified class 'oracle.jdbc.driver.OracleDrive' could not be loaded`", then the issue is with the respective drivers.

To troubleshoot this issue, add the corresponding database drivers in the Administration Console and Identity Server respectively. Restart the Administration Console and Identity Server.

Error 3:

While testing a data source connection, if the error message displays: `"Exception during pool initialization"`, then the issue can be with the incorrect SID specified in the database URL or due to incorrect port or hostname.

To troubleshoot this issue, check the error logs for any data source connection issues:

- Linux: `/opt/novell/nam/logs/adminconsole/tomcat/catalina.out`
- Windows: `\Program Files (x86)\Novell\Tomcat\logs\stdout.log`

Regex Replace Error Message

While performing a regex replace function, if you specify /a, then, the following error message displays:

- On Firefox browser: `Incorrect syntax : invalid regular expression flag a`
- On Chrome browser: `Incorrect syntax : Invalid flags supplied to RegExp constructor 'a'`

To troubleshoot this issue, ensure that you specify a correct regular expression. For example: `i` (case insensitive), `g` (global search) are valid flags. A regular expression has the following format:

`/pattern/modifiers`. Example: var patt = `/abc/i.` Where, `/abc/i` is a regular expression and `abc` is a pattern (used in the search). `i` is a modifier (modifies the search to be case-insensitive).

Using Log Files for Troubleshooting

The following sections provides information about how to use log files for troubleshooting problems:

- "Sample Authentication Traces" on page 1025
- "Understanding Policy Evaluation Traces" on page 1028

Sample Authentication Traces

An authentication trace is logged to the `catalina.out` file of the Identity Server that authenticates the user. If the Access Gateway initiates the authentication because of a user request to a protected resource, the Embedded Service Provider log file of the Access Gateway also contains entries for the authentication sequence. Identity Server logging must be enabled to produce authentication traces (see "Configuring Logging for Identity Server" on page 852).

This section describes the following types of authentication traces:

- "Direct Authentication Request to the Identity Server" on page 1025
- "Protected Resource Authentication Trace" on page 1027

Direct Authentication Request to the Identity Server

The following trace is an example of a user logging directly into the Identity Server to access the end user portal. The log entries are numbered, so that they can be described.

```
1. <amLogEntry> 2009-06-14T17:14:30Z INFO NIDS Application: AM#500105015:
AMDEVICEID#9921459858EAAC29: AMAUTHID#F35A3C7AD7F2EEDEB3D17F99EC3F39D1: Processing
login request with TARGET = http://10.10.15.19:8080/nidp/app, saved TARGET = . </
amLogEntry>

2. <amLogEntry> 2009-06-14T17:14:30Z INFO NIDS Application: AM#500105009:
AMDEVICEID#9921459858EAAC29: AMAUTHID#F35A3C7AD7F2EEDEB3D17F99EC3F39D1: Executing
contract Name/Password - Form. </amLogEntry>

3. <amLogEntry> 2009-06-14T17:14:30Z INFO NIDS Application: AM#500105010:
AMDEVICEID#9921459858EAAC29: AMAUTHID#F35A3C7AD7F2EEDEB3D17F99EC3F39D1: Contract
Name/Password - Form requires additional interaction. </amLogEntry>

4. <amLogEntry> 2009-06-14T17:14:39Z INFO NIDS Application: AM#500105015:
AMDEVICEID#9921459858EAAC29: AMAUTHID#F35A3C7AD7F2EEDEB3D17F99EC3F39D1: Processing
login request with TARGET = http://10.10.15.19:8080/nidp/app, saved TARGET = http:/
/10.10.15.19:8080/nidp/app. </amLogEntry>

5. <amLogEntry> 2009-06-14T17:14:39Z INFO NIDS Application: AM#500105009:
AMDEVICEID#9921459858EAAC29: AMAUTHID#F35A3C7AD7F2EEDEB3D17F99EC3F39D1: Executing
contract Name/Password - Form. </amLogEntry>

6. <amLogEntry> 2009-06-14T17:14:39Z INFO NIDS Application: AM#500105014:
AMDEVICEID#9921459858EAAC29: AMAUTHID#F35A3C7AD7F2EEDEB3D17F99EC3F39D1: Attempting
to authenticate user cn=bcf,o=novell with provided credentials. </amLogEntry>

7. <amLogEntry> 2009-06-14T17:14:39Z WARNING NIDS Application: Event Id: 3014666,
Target: cn=bcf,o=novell, Sub-Target: F35A3C7AD7F2EEDEB3D17F99EC3F39D1, Note 1:
Local, Note 2: This Identity Provider, Note 3: name/password/uri, Numeric 1: 0 </
amLogEntry>

8. <amLogEntry> 2009-06-14T17:14:39Z WARNING NIDS Application: Event Id: 3015456,
Note 1: F35A3C7AD7F2EEDEB3D17F99EC3F39D1, Note 2: Manager, Note 3:
Document=(ou=xpemlPEP,ou=mastercdn,ou=Content
PublisherContainer,ou=Partition,ou=PartitionsContainer,ou=VCDN_Root,ou=accessManag
erContainer,o=novell:romaContentCollectionXMLDoc),Policy=(Manager),Rule=(1::RuleID
_1181251958207),Action=(AddRole::ActionID_1181252224665), Numeric 1: 0 </
amLogEntry>

9. <amLogEntry> 2009-06-14T17:14:39Z WARNING NIDS Application: Event Id: 3015456,
```

```
Note 1: F35A3C7AD7F2EEDEB3D17F99EC3F39D1, Note 2: authenticated, Note 3: system-
generated-action, Numeric 1: 0 </amLogEntry>

10. <amLogEntry> 2009-06-14T17:14:39Z INFO NIDS Application: AM#500199050:
AMDEVICEID#9921459858EAAC29: AMAUTHID#F35A3C7AD7F2EEDEB3D17F99EC3F39D1: IDP
RolesPep.evaluate(), policy trace:
   ~~RL~1~~~~Rule Count: 1~~Success(67)
   ~~RU~RuleID_1181251958207~Manager~DNF~~1:1~~Success(67)
   ~~CS~1~~ANDs~~1~~True(69)
   ~~CO~1~LdapGroup(6645):no-param:hidden-value:~ldap-group-is-member-
of~SelectedLdapGroup(66455):hidden-param:hidden-value:~~~True(69)
   ~~PA~ActionID_1181252224665~~AddRole~Manager~~~Success(0)
   ~~PC~ActionID_1181252224665~~Document=(ou=xpemlPEP,ou=mastercdn,
ou=ContentPublisherContainer,ou=Partition,ou=PartitionsContainer,ou=VCDN_Root,ou=a
ccessManagerContainer,o=novell:romaContentCollectionXMLDoc),Policy=(Manager),Rule=
(1::RuleID_1181251958207),Action=(AddRole::ActionID_1181252224665)~AdditionalRole(
6601):unknown():Manager:~~~Success(0)
 </amLogEntry>

11. <amLogEntry> 2009-06-14T17:14:39Z INFO NIDS Application: AM#500105013:
AMDEVICEID#9921459858EAAC29: AMAUTHID#F35A3C7AD7F2EEDEB3D17F99EC3F39D1:
Authenticated user cn=bcf,o=novell in User Store Local Directory with roles
Manager,authenticated. </amLogEntry>

12. <amLogEntry> 2009-06-14T17:14:39Z INFO NIDS Application: AM#500105017:
AMDEVICEID#9921459858EAAC29: AMAUTHID#F35A3C7AD7F2EEDEB3D17F99EC3F39D1: nLogin
succeeded, redirecting to http://10.10.15.19:8080/nidp/app. </amLogEntry>
```

Table 26-3 *Log Entry Descriptions for an Authentication Trace from an Identity Server*

Entry	Description
1	Indicates that a login request is in process. This is the first entry for a login request. The requester, even though login has not been successful, is assigned an authentication ID. You can use this ID to find the log entries related to this user. The entry also specifies the URL of the requested resource, in this case the /nidp/app resource called the End User Portal. The saved TARGET message does not contain a value, so this step will be repeated.
2	Specifies the contract that is being used to perform the login.
3	Indicates that the contract requires interaction with the user.
4	Indicates that the a login request is in process. The saved TARGET message contains a value, so the required information has been gathered to start the authentication request. The AM# correlation tag is `AM#500105015`, which is the same value as the first log entry.
5	Indicates that an exchange is occurring between the client and the Identity Server to obtain the required credentials. Each contract requires a different exchange. The `AM#` correlation tag is AM#500105009, which is the same value as the second log entry.
6	Provides the DN of the user attempting to log in and indicates that the user's credentials are being sent to the LDAP server for verification.
7	Provides information about an auditing event. If you have not enabled auditing or you have not selected the login events, this entry does not appear in your log file. This event contains information about who is logging in and the contract that is being used.
8	Provides information about an auditing event. If you have not enabled auditing or you have not selected the login events, this entry does not appear in your log file. This event contains information about the Manager policy that is evaluated during login.

Entry	Description
9	Provides information about an auditing event. If you have not enabled auditing or you have not selected the login events, this entry does not appear in your log file.
10	Contains the entry for processing a Role policy. When a user logs in, all Role policies are evaluated and the user is assigned any roles that the user has the qualifications for. For more information, see “Understanding Policy Evaluation Traces” on page 1028.
11	Contains a summary of who logged in from which user store and the names of the Role policies that successfully assigned roles to the user.
12	Contains the final results of the login, with the URL that the request is redirected to.

Protected Resource Authentication Trace

When a protected resource is configured to require authentication, both the Identity Server and the Embedded Service Provider of the Access Gateway generate log entries for the process. The following sections explain how to correlate the entries from the logs.

- “Entries from an Identity Server Log” on page 1027
- “Entries from an Access Gateway Log” on page 1028
- “Correlating the Log Entries between the Identity Server and the Access Gateway” on page 1028

Entries from an Identity Server Log

```
<amLogEntry> 2009-07-31T17:36:39Z INFO NIDS Application: AM#500105016:
AMDEVICEID#AA257DA77ED48DB0: AMAUTHID#83778AE09DCA5A35B57842D754A60D67: Processing
login resulting from Service Provider authentication request. </amLogEntry>

<amLogEntry> 2009-07-31T17:36:39Z INFO NIDS Application: AM#500105009:
AMDEVICEID#AA257DA77ED48DB0: AMAUTHID#83778AE09DCA5A35B57842D754A60D67: Executing
contract Name/Password - Form. </amLogEntry>

<amLogEntry> 2009-07-31T17:36:39Z INFO NIDS Application: AM#500105010:
AMDEVICEID#AA257DA77ED48DB0: AMAUTHID#83778AE09DCA5A35B57842D754A60D67: Contract
Name/Password - Form requires additional interaction. </amLogEntry>

<amLogEntry> 2009-07-31T17:36:49Z INFO NIDS Application: AM#500105016:
AMDEVICEID#AA257DA77ED48DB0: AMAUTHID#83778AE09DCA5A35B57842D754A60D67: Processing
login resulting from Service Provider authentication request. </amLogEntry>

<amLogEntry> 2009-07-31T17:36:49Z INFO NIDS Application: AM#500105009:
AMDEVICEID#AA257DA77ED48DB0: AMAUTHID#83778AE09DCA5A35B57842D754A60D67: Executing
contract Name/Password - Form. </amLogEntry>

<amLogEntry> 2009-07-31T17:36:49Z INFO NIDS Application: AM#500105014:
AMDEVICEID#AA257DA77ED48DB0: AMAUTHID#83778AE09DCA5A35B57842D754A60D67: Attempting
to authenticate user cn=admin,o=novell with provided credentials. </amLogEntry>

<amLogEntry> 2009-07-31T17:36:49Z INFO NIDS Application: AM#500105012:
AMDEVICEID#AA257DA77ED48DB0: AMAUTHID#83778AE09DCA5A35B57842D754A60D67:
Authenticated user cn=admin,o=novell in User Store Internal with no roles. </
amLogEntry>
```

```
<amLogEntry> 2009-07-31T17:36:49Z INFO NIDS Application: AM#500105018:
AMDEVICEID#AA257DA77ED48DB0: AMAUTHID#83778AE09DCA5A35B57842D754A60D67: Responding
to AuthnRequest with artifact AAMoz+rm2jQjDSHjea8U9zm3Td/U2ax0YZCo/
qBNool8WkZiTCt7N7Jx </amLogEntry>

<amLogEntry> 2009-07-31T17:36:49Z INFO NIDS Application: AM#500105019:
AMDEVICEID#AA257DA77ED48DB0: AMAUTHID#C2D8D52704918AF2D5D62F6EDC2FFAC6: Sending
AuthnResponse in response to artifact AAMoz+rm2jQjDSHjea8U9zm3Td/U2ax0YZCo/
qBNool8WkZiTCt7N7Jx </amLogEntry>
```

Entries from an Access Gateway Log

```
<amLogEntry> 2009-07-31T17:35:05Z INFO NIDS Application: AM#500105005:
AMDEVICEID#esp-2FA73CE1A376FD91: AMAUTHID#C6D119FD93EEBBEBEC50BEB27B9E2832:
Processing proxy request for login using contract name/password/uri and return url
http://jwilson.provo.novell.com/ </amLogEntry>

<amLogEntry> 2009-07-31T17:35:05Z INFO NIDS Application: AM#500105015:
AMDEVICEID#esp-2FA73CE1A376FD91: AMAUTHID#C6D119FD93EEBBEBEC50BEB27B9E2832:
Processing login request with TARGET = http://jwilson.provo.novell.com/, saved
TARGET = . </amLogEntry>

<amLogEntry> 2009-07-31T17:35:05Z INFO NIDS Application: AM#500105009:
AMDEVICEID#esp-2FA73CE1A376FD91: AMAUTHID#C6D119FD93EEBBEBEC50BEB27B9E2832:
Executing contract IDP Select. </amLogEntry>

<amLogEntry> 2009-07-31T17:35:05Z INFO NIDS Application: AM#500105010:
AMDEVICEID#esp-2FA73CE1A376FD91: AMAUTHID#C6D119FD93EEBBEBEC50BEB27B9E2832:
Contract IDP Select requires additional interaction. </amLogEntry>

<amLogEntry> 2009-07-31T17:35:15Z INFO NIDS Application: AM#500105020:
AMDEVICEID#esp-2FA73CE1A376FD91: AMAUTHID#C6D119FD93EEBBEBEC50BEB27B9E2832:
Received and processing artifact from IDP - AAMoz+rm2jQjDSHjea8U9zm3Td/U2ax0YZCo/
qBNool8WkZiTCt7N7Jx </amLogEntry>

<amLogEntry> 2009-07-31T17:35:15Z INFO NIDS Application: AM#500105021:
AMDEVICEID#esp-2FA73CE1A376FD91: AMAUTHID#C6D119FD93EEBBEBEC50BEB27B9E2832:
Sending artifact AAMoz+rm2jQjDSHjea8U9zm3Td/U2ax0YZCo/qBNool8WkZiTCt7N7Jx to URL
http://jwilson1.provo.novell.com:8080/nidp/idff/soap at IDP </amLogEntry>
```

Correlating the Log Entries between the Identity Server and the Access Gateway

You can see that these two trace sequences are for the same authentication request because the artifact (`AAMoz+rm2jQjDSHjea8U9zm3Td/U2ax0YZCo/qBNool8WkZiTCt7N7Jx`) that is exchanged is the same. You can use the `AMAUTHID` in each file to search for other requests that this user has made.

To associate a distinguished name with the `AMAUTHID`, use the `catalina.out` file of the Identity Server. Event `AM#500105014` contains the DN of the user.

Understanding Policy Evaluation Traces

- “Format” on page 1029
- “Policy Result Values” on page 1035
- “Role Assignment Traces” on page 1036
- “Identity Injection Traces” on page 1037

- “Authorization Traces” on page 1039
- “Form Fill Traces” on page 1041

Format

A policy log entry starts with the standard log entry elements: <amLogEntry> followed by the correlation tags.

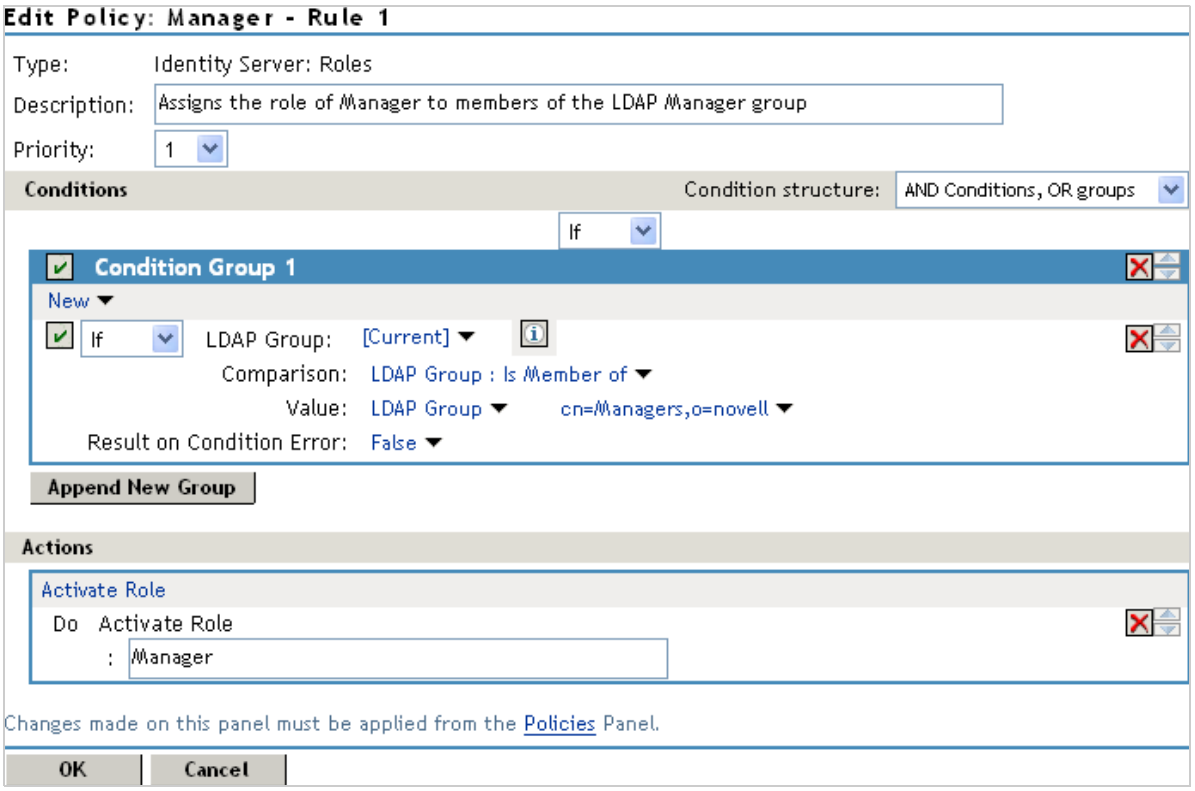
(For information about correlation tags, see “Understanding the Correlation Tags in the Log Files” on page 850.)

The following log entry is a trace of an evaluation of a Role policy:

```
<amLogEntry> 2009-06-07T21:40:25Z INFO NIDS Application: AM#500199050:
AMDEVICEID#9921459858EAAC29: AMAUTHID#503EFFA4BC21ACA307796EC7D96E5532: IDP
RolesPep.evaluate(), policy trace:
   ~~RL~0~~~~Rule Count: 1~~Success(67)
   ~~RU~RuleID_1181251958207~Manager~DNF~~1:1~~Success(67)
   ~~CS~1~~ANDs~~1~~True(69)
   ~~CO~1~LdapGroup(6645):no-param:hidden-value:~ldap-group-is-member-
of~SelectedLdapGroup(66455):hidden-param:hidden-value:~~~True(69)
   ~~PA~ActionID_1181252224665~~AddRole~Manager~~~Success(0)
   ~~PC~ActionID_1181252224665~~Document=(ou=xpemlPEP,ou=mastercdn,
ou=ContentPublisherContainer,ou=Partition,ou=PartitionsContainer,ou=VCDN_Root,ou=a
ccessManagerContainer,o=novell:romaContentCollectionXMLDoc),Policy=(Manager),Rule=
(1::RuleID_1181251958207),Action=(AddRole::ActionID_1181252224665)~AdditionalRole(
6601):unknown():Manager:~~~Success(0)
 </amLogEntry>
```

The Role policy evaluated in this entry has the following definition:

Figure 26-9 *Manager Policy Definition*



The following sections use this policy and its trace to explain the information contained within each line of a policy trace. The policy trace part of the entry starts with a `policy trace:`, which is followed by one or more of the following types:

- RL - Rule List Evaluation Result (page 1030)
- RU - "Rule Evaluation Result" on page 1031
- CS - Condition Set Evaluation Result (page 1032)
- CO - Condition Evaluation Result (page 1032)
- PA - Policy Action Initiation (page 1034)
- PC - Policy Action Completion (page 1034)

Elements within a type are separated from each other with the tilde (~) character. If an element does not have a value, no value is inserted, which results in two or more tildes between values. Two tildes means one element didn't have a value, three tildes means that two elements didn't have values, and so forth.

Rule List Evaluation Result

An RL trace has the following fields:

```
~<RuleListID>~~~~<RuleCount>~~<Result>
```

A RL trace looks similar to the following:

```
~~RL~1~~~~Rule Count: 1~~Success(67)
```

Table 26-4 describes the fields found in an RL trace.

Table 26-4 Fields in a Rule List Trace

Element	Description
<RuleListID>	The identifier assigned to the rule list. In the sample RL trace, this is 1.
<RuleCount>	The number of rules defined for the policy. In the sample RL trace, this is `Rule Count: 1`, indicating that there is one rule in the policy.
<Result>	A string followed by a number that specifies the result of the evaluation. See "Policy Result Values" on page 1035. In the sample RL trace, this is `Success(67)`, indicating success.

Rule Evaluation Result

An RU trace has the following fields:

```
~<RuleID>~<ParentPolicyName>~<ConditionSetJoinType>~~~<ConditionSetCount:
ActionCount>~~~<Result>
```

An RU trace looks similar to the following:

```
~~RU~RuleID_1181251958207~Manager~DNF~~1:1~~Success(67)
```

Table 26-5 describes the fields of a Rule Evaluation Result trace.

Table 26-5 Fields in a Rule Evaluation Result Trace

Element	Description
<RuleID>	The identifier assigned to the rule. In this sample RU trace, this element is set to `RuleID_1181251958207`.
<ParentPolicyName>	The name of the parent policy to which the rule is assigned. In this sample RU trace, this element is set to `Manager`.
<ConditionSetJoinType>	The type of joining that occurs between conditions and condition sets. It is set to one of the following: ◆ **CNF:** Indicates that sets are ANDed and conditions within a condition group are ORed. ◆ **DNF:** Indicates that sets are ORed and conditions within a condition group are ANDed. In the sample RU trace, this element is set to `DNF`.
<ConditionSetCount:ActionCount>	The number of condition sets and actions defined for this rule. In the sample RU trace, this is 1:1, for one condition set and one action.

Element	Description
<Result>	A string followed by a number that specifies the result of the evaluation. See “Policy Result Values” on page 1035. In the sample RU trace, this is Success(67), indicating that the rule was successfully evaluated.

Condition Set Evaluation Result

A CS trace has the following fields

```
~<ConditionSetID>~<JoinType>~<NOT>~<ConditionCount>~~~<Result>
```

A CS trace looks similar to the following:

```
~~CS~1~~ANDs~~1~~True(69)
```

Table 26-6 describes the fields in a Condition Set trace.

Table 26-6 *Fields in a Condition Set Trace*

Element	Description
<ConditionSetID>	The identifier assigned to the condition set. Rules can have multiple condition sets. In this sample CS trace, this is 1, for the first and only condition set defined for the rule.
<JoinType>	Specifies how the condition results are combined, if there are multiple condition sets. Possible values include ANDs and ORs. In this sample CS trace, this is ANDs.
<NOT>	The string NOT if the result was negated prior to reporting; otherwise the field has no value. This is the **If Not** option when creating a condition group. In the sample CS trace, the condition group was not negated, therefore the field is not present.
<ConditionCount>	The number of conditions defined in the condition group. In the sample CS trace, this element has the value of 1.
<Result>	A string followed by a number that specifies the result of the evaluation. See “Policy Result Values” on page 1035. In the sample CS trace, this is True (69), indicating that the condition evaluated to True.

Condition Evaluation Result

A CO trace has the following fields:

```
~<ConditionID>~<LHSOperand>~<Operator>~<RHSOperand>~<NOT>~<Result>[~<ResultOnError
>]
```

A CO trace looks similar to the following:

```
~~CO~1~LdapGroup(6645):no-param:hidden-value:~ldap-group-is-member-
of~SelectedLdapGroup(66455):hidden-param:hidden-value:~~~True(69)
```

Table 26-7 describes the fields in a Condition trace.

Table 26-7 *Fields in a Condition Trace*

Element	Description
<ConditionID>	The identifier assigned to the conditions in the condition group. The first condition is assigned 1. In the sample CO trace, this is 1.
<LHSOperand>	The enumerative value and parameter list of the left operand. It is the first value specified for the comparison and has the following format: <Condition Name(Data ID)>: <Parameter> : <Value> The Condition Name is the string assigned to the condition type specified in the policy. The Data ID is a numerical value assigned to the condition type. <Parameter> contains one of the following strings: ◆ no-param when no parameters are specified for the operand, followed by a colon, followed by one of the following: the value, no-value, or hidden-value when the value contains sensitive information. ◆ hidden-param followed by a colon, and then hidden-value. This string is used when both the parameter and its value contain sensitive information. In the sample CO trace, this is LdapGroup(6645):no-param:hidden-value. LdapGroup is the string for the LDAP Group condition. The policy specified **[Current]**, so no parameters were specified. The groups that the user belongs to are considered sensitive data, so the log file displays hidden-value for the names of the groups.
<Operator>	The display name of the comparison operator. In the sample CO trace, this is ldap-group-is-member-of. In the policy, this is displayed as **LDAP Group: Is Member of**.
<RHSOperand>	The enumerative value and parameter list of the right operand. It is the second value specified for the comparison and has the same format as the <LHSOperand>. In the sample CO trace, this is SelectedLdapGroup(66455):hidden-param:hidden-value. The actual policy specifies LDAP Group as the parameter, and the value is the DN of the group.
<NOT>	The string NOT if the result was negated prior to reporting; otherwise the field has no value. This is the **If Not** option when creating a condition. In the sample CO trace, this condition result was not negated, therefore the field is represented by a tilde.
<Result>	A string followed by a number that specifies the result of the comparison. See "Policy Result Values" on page 1035. In the sample CO trace, this is True (69), indicating that the condition evaluated to True—the user is a member of the specified LDAP group.
<ResultOnError>	A string describing the error that occurred. This is an optional field that only appears when the condition evaluation results in an error. The sample CO trace did not result in an error, so it has no string.

Policy Action Initiation

A PA trace has the following fields:

```
~<ActionID>~<TraceString1>~<TraceString2>~<TraceString3>~<Result>
```

A PA trace looks similar to the following:

```
~~PA~ActionID_1181252224665~~AddRole~Manager~~~Success(0)
```

Table 26-8 describes the fields in a Policy Action trace.

Table 26-8 *Fields in a Policy Action Trace*

Element	Description
`<ActionID>`	The identifier assigned to the action. In the sample PA trace, this is `ActionID_1181252224665`.
`<TraceString1>`	The message specified with the action. In the sample PA trace, this is `AddRole`.
`<TraceString2>`	The second part of the specified message. In the sample PA trace, this is `Manager`.
`<TraceString3>`	The third part of the specified message. In the sample PA trace, this field has no value and is not present.
`<Result>`	A string followed by a number that specifies the result of the assigning the action. See “Policy Result Values” on page 1035. In the sample PA trace, this is `Success(0),` which indicates that the action of assigning the Manager role to the user was successful.

Policy Action Completion

A PC trace has the following fields

```
~<ActionID>~<ActionName>~<ActionParmeters>~~~<Result>[~<ActionError>]
```

A PC trace looks similar to the following:

```
~~PC~ActionID_1181252224665~~Document=(ou=xpemlPEP,ou=mastercdn,
ou=ContentPublisherContainer,ou=Partition,ou=PartitionsContainer,ou=VCDN_Root,ou=a
ccessManagerContainer,o=novell:romaContentCollectionXMLDoc),Policy=(Manager),Rule=
(1::RuleID_1181251958207),Action=(AddRole::ActionID_1181252224665)~AdditionalRole(
6601):unknown():Manager:~~~Success(0)
```

Table 26-9 describes the fields in a Policy Action Completion trace.

Table 26-9 *Fields in a Policy Action Completion Trace*

Element	Description
`<ActionID>`	The ID assigned to the action. In the sample PC trace, this is `ActionID_1181252224665`.

Element	Description
<ActionName>	The fully distinguished name of the action. In the sample PC trace, the action has the following parts in its name: ◆ Document=(ou=xpemlPEP,ou=mastercdn, ou=ContentPublisherContainer,ou=Partition,ou=PartitionsContainer,ou=VCDN_Root,ou=accessManagerContainer,o=novell:romaContentCollectionXMLDoc) ◆ Policy=(Manager) ◆ Rule=(1::RuleID_1181251958207) ◆ Action=(AddRole::ActionID_1181252224665)
<ActionParmeters>	A list of the action parameters passed to the action handler. In this sample PC trace, the Role policy has an action and a parameter. The value of this element is AdditionalRole(6601):unknown(): Manager:
<Result>	A string followed by a number that specifies the result. See “Policy Result Values” on page 1035. In the sample PC trace, this is Success(0)and indicates success.
<ActionError>	A string describing the error that occurred when invoking the action. This is an optional field that only appears when the Result field contains an error code. The sample PC trace did not result in an error, so it has no string.

Policy Result Values

The last field in a trace string is the <result> field. Table 26-10 lists the possible values:

Table 26-10 *Result Values from Policy Traces*

Value	Name	Description
0	Success	The policy evaluation was successful.
1	Error: No memory	The system is out of memory.
2	Error: Bad data	The data sent for evaluation is invalid.
3	Error: Configuration initialization	An error was detected during the policy configuration processing.
4	Error: General failure	An error was detected during policy processing.
5	Pending	The policy processing is in progress.
64	Permit	The rule produced a Permit action.
65	Deny	The rule produced a Deny action.
66	Obligation	The rule triggered an obligation, indicating that additional processing is required. Identity Injection policies trigger obligations.
67	No action	The rule did not initiate any action.

Value	Name	Description
68	Condition false	The condition evaluated to False.
69	Condition true	The condition evaluated to True.
70	Condition unknown	Condition input was not available, so the results are unknown.
71	Cancel	The current operation has been canceled.
72	Error: Interface unavailable	The current operation is unavailable.
73	Error: Data unavailable	The data required for evaluation was unavailable.
74	Error: Illegal state	Processing error; report it to Novell® Support.

Role Assignment Traces

The following sections walk you through a few sample role traces. When you understand these traces, you should be able to understand any role trace.

- "When the User Is Assigned Roles" on page 1036
- "When the Role Policy Is Not Enabled" on page 1037
- "When an Authorization Policy Uses a Role" on page 1037

When the User Is Assigned Roles

Roles are assigned at authentication, so this type of trace is found in the `catalina.out` file of the Identity Server. This is a trace of a user who does not match the requirements to be assigned the Manager Role (for a definition of this Role policy, see Figure 26-9 on page 1030).

```
<amLogEntry> 2009-06-11T15:38:38Z INFO NIDS Application: AM#500199050:
AMDEVICEID#9921459858EAAC29: AMAUTHID#0CE611AAE4D0301F26DD4865476BDA1 4: IDP
RolesPep.evaluate(), policy trace:
   ~~RL~0~~~~Rule Count: 1~~Success(67)
   ~~RU~RuleID_1181251958207~Manager~DNF~~1:1~~Success(67)
   ~~CS~1~~ANDs~~1~~False(68)
   ~~CO~1~LdapGroup(6645):no-param:hidden-value:~ldap-group-is-member-
of~SelectedLdapGroup(66455):hidden-param:hidden-value:~~~False(68)
</amLogEntry>
```

This trace describes the following about the policy.

1. The RL trace indicates that the policy has one rule and that the policy evaluated without error.
2. The RU trace indicates that the rule (`RuleID_1181251958207`) has one condition and one action and that the rule evaluated without error.
3. The CS trace indicates that the condition set evaluated to False (the user logging in does not match the conditions of the set).
4. The CO trace indicates that the condition evaluated to False (the user logging in does not match the condition).

When you are troubleshooting why a user is not granted access to a resource that uses a role in its Authentication policy, the first step should be to look at the Identity Server file and determine whether the user was assigned the role. In this trace, you can see that the user was not assigned the role. To fix this problem, you can either change the conditions of the Role policy to match the user or change the user's information so that the user matches the existing condition in the Role policy.

When the Role Policy Is Not Enabled

Sometimes a Role policy is created, but the Role policy is not enabled for the Identity Server. When this happens, the trace looks similar to the following:

```
<amLogEntry> 2009-06-11T16:06:03Z INFO NIDS Application: AM#500199050:
AMDEVICEID#9921459858EAAC29: AMAUTHID#FDE680ABE320B682038947EA5F59D6B F: IDP
RolesPep.evaluate(), policy trace:
   ~~RL~0~~~~Rule Count: 0~~Success(67)
 </amLogEntry>
```

When you see Role policy traces that contain only the RL trace line, you need to enable the Role policy.

When an Authorization Policy Uses a Role

When a user requests access to a resource that has an Authorization policy that uses a role, the user is checked for the role assignment. The trace of this evaluation is in the ESP log file of the Access Gateway that is processing the request. Such a trace looks similar to the following:

```
<amLogEntry> 2009-07-13T22:13:29Z INFO NIDS Application: AM#501102050:
AMDEVICEID#esp-51A474B83BFDDF4F: AMAUTHID#4538DB6F6E2A237FDE674F0C6E1 6DCEC:
PolicyID#N748097P-3507-3KP7-4241-410PN4152094: NXPESID#1718: AGAuthorization
Policy Trace:
   ~~RL~1~~~~Rule Count: 1~~Success(0)
   ~~RU~RuleID_1182876316974~Allow_Sales~DNF~~1:1~~Success(0)
   ~~CS~1~~ANDs~NOT~1~~True(69)
   ~~CO~1~CurrentRoles(6660):no-param:authenticated~com.novell.nxpe.
condition.NxpeOperator@string-substring~SelectedRole(6661):hidden-param:hidden-
value:~~~False(68)
   ~~PA~1~~Deny Access Messasge~Sorry, you must work in sales today.~~~Success(0)
   ~~PC~1~~Document=(ou=xpemlPEP,ou=mastercdn,ou=ContentPublisherCon
tainer,ou=Partition,ou=PartitionsContainer,ou=VCDN_Root,ou=accessManagerContainer,
o=novell:romaContentCollectionXMLDoc),Policy=(Allow_Sales),Rule=(1::RuleID_1182876
316974),Action=(Deny::1)~~~~Success(0)
</amLogEntry>
```

This trace is for a Deny policy that denies access if the user has not been assigned the Sales role. The CO line indicates that the condition is looking for a role and that the user did not match the condition.

The CS line indicates that the condition is a negative condition, meaning that the user matches the condition set when the user does not match the condition. This is the case for this user, so the condition set evaluates to True, and the action is then applied.

The PA line describes the action that was applied.

Identity Injection Traces

The following traces explain what to look for in an Identity Injection policy that injects an authorization header:

- “When the User Has Authenticated” on page 1038
- “When the User Has Not Authenticated” on page 1039

When the User Has Authenticated

The following trace is for an Identity Injection policy that successfully inserts an authentication header. The policy inserts LDAP credentials for the user's name and password. The Access Gateway injects the information, so the trace for this type of policy is in the ESP log file of the Access Gateway.

```
<amLogEntry> 2009-06-11T19:02:44Z INFO NIDS Application: AM#501103050:
AMDEVICEID#esp-534FD0D0E32FE4BD: AMAUTHID#61D5D5B3FF98156F8E4F2875981D 4A6E:
PolicyID#51N4214K-74L1-491L-7190-2M9K04K21393: NXPESID#726: AGIdentityInjection
Policy Trace:
   ~~RL~0~~~~Rule Count: 1~~Success(67)
   ~~RU~RuleID_1181251426062~basic_auth_ii~DNF~~0:1~~Success(67)
   ~~PA~ActionID_1181251427701~~Inject Auth Header~uid~uid(1):
CredentialProfile(7010:):NEPXurn~3Anovell~3Acredentialprofile~3A2005-
03~2Fcp~3ASecrets~2Fcp~3ASecret~2Fcp~3AEntry~40~40~40~40WSCQSSToken~40~40~40~40~2F
cp~3ASecrets~2Fcp~3ASecret~5Bcp~3AName~3D~22LDAPCredentials~22~5D~2Fcp~3AEntry~5Bc
p~3AName~3D~22UserName~22~5D:~Ok~Success(0)
   ~~PA~ActionID_1181251427701~~Inject Auth Header~password~pwd(1):
CredentialProfile(7010:):NEPXurn~3Anovell~3Acredentialprofile~3A2005-
03~2Fcp~3ASecrets~2Fcp~3ASecret~2Fcp~3AEntry~40~40~40~40WSCQSSToken~40~40~40~40~2F
cp~3ASecrets~2Fcp~3ASecret~5Bcp~3AName~3D~22LDAPCredentials~22~5D~2Fcp~3AEntry~5Bc
p~3AName~3D~22UserPassword~22~5D:~Ok~Success
(0)
   ~~PC~ActionID_1181251427701~~Document=(ou=xpemlPEP,ou=mastercdn,
ou=ContentPublisherContainer,ou=Partition,ou=PartitionsContainer,ou=VCDN_Root,ou=a
ccessManagerContainer,o=novell:romaContentCollectionXMLDoc),Policy=(basic_auth_ii)
,Rule=(1::RuleID_1181251426062),Action=(InjectAuthHeader::ActionID_1181251427701)~
~~~Success(0)
 </amLogEntry>

<amLogEntry> 2009-06-11T19:02:44Z INFO NIDS Application: AM#501101021:
AMDEVICEID#esp-534FD0D0E32FE4BD: AMAUTHID#61D5D5B3FF98156F8E4F2875981D 4A6E:
PolicyID#51N4214K-74L1-491L-7190-2M9K04K21393: NXPESID#726: Response sent: Status
- success </amLogEntry>
```

Each identity injection policy generates two log entries. The first entry indicates whether the policy could successfully retrieve the information and inject it into the header. The second entry specifies whether the response is successfully sent to the Web server.

This first log entry describes the following about this policy:

1. In the correlation tags (AM... tags), notice the ID assigned to the authenticated user making the request (`AMAUTHID#61D5D5B3FF98156F8E4F2875981D4A6E`).
2. After the correlation tags, the trace specifies the ID of the policy (`51N4214K-74L1-491L-7190-2M9K04K21393`).
3. The RU trace indicates that the policy name is basic_auth_ii, that the policy has no conditions, and that the policy has one action rule.
4. The first PA trace indicates that the uid (called LDAP User Name in the UI) of the Credential Profile has been successfully retrieved.
5. The second PA trace indicates that the password of the Credential Profile has been successfully retrieved.
6. The PC trace indicates that these items have been successfully injected into the header.

You can use the user's ID and the policy ID to find log entry that traces the response to the Web server. The second log entry indicates that the response was successfully sent to the Web server.

When the User Has Not Authenticated

If the user has not authenticated and therefore has no authentication credentials, the trace for an Identity Injection policy with an authentication header looks similar to the following:

```
<amLogEntry> 2009-06-11T20:16:51Z INFO NIDS Application: AM#501103050:
AMDEVICEID#esp-534FD0D0E32FE4BD: PolicyID#OL8659PL-0K69-0N0N-0845-5PN113KM3842:
NXPESID#2539: AGIdentityInjection Policy Trace:
   ~~RL~0~~~~Rule Count: 1~~Success(67)
   ~~RU~RuleID_1181251426062~basic_auth_ii~DNF~~0:1~~Success(67)
   ~~PA~ActionID_1181251427701~~Inject Auth Header~uid~uid(1):
CredentialProfile(7010:):NEPXurn~3Anovell~3Acredentialprofile~3A2005-
03~2Fcp~3ASecrets~2Fcp~3ASecret~2Fcp~3AEntry~40~40~40~40WSCQSSToken~40~40~40~40~2F
cp~3ASecrets~2Fcp~3ASecret~5Bcp~3AName~3D~22LDAPCredentials~22~5D~2Fcp~3AEntry~5Bc
p~3AName~3D~22UserName~22~5D:~Ok~Success(0)
   ~~PA~ActionID_1181251427701~~Inject Auth
Header~password~pwd(1):CredentialProfile(7010:):NEPXurn~3Anovell~3Acredentialprofi
le~3A2005-03~2Fcp~3ASecrets~2Fcp~3ASecret~2Fcp~3AEntry
~40~40~40~40WSCQSSToken~40~40~40~40~2Fcp~3ASecrets~2Fcp~3ASecret~5Bcp~3AName~3D~22
LDAPCredentials~22~5D~2Fcp~3AEntry~5Bcp~3AName~3D~22UserPassword~22~5D:~Ok~Success
(0)
   ~~PC~ActionID_1181251427701~~Document=(ou=xpemlPEP,ou=mastercdn,
ou=ContentPublisherContainer,ou=Partition,ou=PartitionsContainer,ou=VCDN_Root,ou=a
ccessManagerContainer,o=novell:romaContentCollectionXMLDoc),Policy=(basic_auth_ii)
,Rule=(1::RuleID_1181251426062),Action=(InjectAuthHeader::ActionID_1181251427701)~
~~~Success(0)
 </amLogEntry>

<amLogEntry> 2009-06-11T20:16:51Z INFO NIDS Application: AM#501101021:
AMDEVICEID#esp-534FD0D0E32FE4BD: PolicyID#OL8659PL-0K69-0N0N-0845-5PN113KM3842:
NXPESID#2539: Response sent: Status - success </amLogEntry>
```

These entries look very similar to the entries for a successful injection of data. This is because injecting NULL data for data that is not available is considered a successful action. The trace displays data unavailable errors only when errors occur retrieving data. The key to determining whether the data was available for injection into an authentication header is to look for the `AMAUTHID` correlation tag in the log entry. The log entries for the `OL8659PL-0K69-0N0N-0845-5PN113KM3842` policy do not contain an `AMAUTHID` correlation tag, which indicates that the user is not logged in.

Authorization Traces

Authorization policies for a protected resource might require a user to be authenticated before the data required by the policy can be obtained, but Authorization policies can be configured to use data that is available without authentication. The following traces show how the log entries for an Authorization policy trace are slightly different when the user is not authenticated.

- “When the Protected Resource Requires Authentication” on page 1040
- “When the Protected Resource Does Not Require Authentication” on page 1041

For a trace of an Authorization policy that uses a role, see “When an Authorization Policy Uses a Role” on page 1037.

When the Protected Resource Requires Authentication

The following is a successful trace of an Authorization policy that requires the user to have the value of Manager in the title attribute. To obtain this data, the user must be authenticated.

The policy contains two rules: a Permit rule if the user has the value of Manager in the title attribute, and a Deny rule that denies all other users. This policy has been assigned to protect an Access Gateway resource.

```
<amLogEntry> 2009-08-02T15:55:05Z INFO NIDS Application: AM#501101050:
AMDEVICEID#esp-2FA73CE1A376FD91: PolicyID#459O8443-N8P5-KO21-68OM-K172P107N4O5:
NXPESID#1743: Evaluating policy </amLogEntry>

<amLogEntry> 2009-08-02T15:55:06Z INFO NIDS Application: AM#501102050:
AMDEVICEID#esp-2FA73CE1A376FD91: AMAUTHID#838976482579AF372C31C4727 4E9CB28:
PolicyID#459O8443-N8P5-KO21-68OM-K172P107N4O5: NXPESID#1743: AGAuthorization
Policy Trace:
   ~~RL~1~~~~Rule Count: 2~~Success(0)
   ~~RU~RuleID_1186068489688~Title_auth~DNF~~1:1~~Success(0)
   ~~CS~1~~ANDs~~1~~True(69)
   ~~CO~1~LdapAttribute(6647):NEPXurn~3Anovell~3Aldap~3A2006-
02~2Fldap~3AUserAttribute~40~40~40~40WSCQLDAPToken~40~40~40~40~2FUserAttribute~5B~
40ldap~3AtargetAttribute~3D~22title~22~5D:hidden-
value:~com.novell.nxpe.condition.NxpeOperator@string-equals~(0):hidden-
param:hidden-value:~~~True(69)
   ~~PA~1~~Permit Access~~~~Success(0)
   ~~PC~1~~Document=(ou=xpemlPEP,ou=mastercdn,ou=ContentPublisher
Container,ou=Partition,ou=PartitionsContainer,ou=VCDN_Root,ou=accessManagerContain
er,o=novell:romaContentCollectionXMLDoc),Policy=(Title_auth),Rule=(1::RuleID_11860
68489688),Action=(Permit::1)~~~~Success(0)
</amLogEntry>

<amLogEntry> 2009-08-02T15:55:06Z INFO NIDS Application: AM#501101021:
AMDEVICEID#esp-2FA73CE1A376FD91: AMAUTHID#838976482579AF372C31C47274E 9CB28:
PolicyID#459O8443-N8P5-KO21-68OM-K172P107N4O5: NXPESID#1743: Response sent: Status
- success </amLogEntry>
```

The first log entry is the request to evaluate the policy. The second log entry is the evaluation of the policy. The third log entry is the response that is returned. These three log entries can be tied together by using the following tags:

AMDEVICEID#esp-2FA73CE1A376FD91: When a policy evaluation request is made, the same ESP processes the request. Even if the Access Gateways are clustered, the policy evaluation request stays with the Access Gateway that initiated the request.

PolicyID#459O8443-N8P5-KO21-68OM-K172P107N4O5: Each policy is assigned a unique ID, and this is the ID assigned to the policy called Title_auth in the Administration Console. To search for all log entries for a policy, use the policy ID. To search for log entries that evaluate the policy, use the policy name.

AMAUTHID#838976482579AF372C31C47274E9CB28: The request to evaluate a policy does not contain the ID of the user the request is being made for, but the log entries for the evaluation and for the response status always contain the ID of an authenticated user. If the policy can be evaluated without the user being authenticated, these entries do not contain the ID of the user. This kind of policy might be assigned to a public resource (no authentication required) and use the time of day condition or day of the week condition for its evaluation criteria. See “When the Protected Resource Does Not Require Authentication” on page 1041.

When the Protected Resource Does Not Require Authentication

The following trace is for an Authorization policy that uses data that is available without authentication. Authorization policies support a number of these conditions, such as Current Date, Current Day of Week, Current Day of Month, Current Time Of Day, Client IP, and the URL conditions. As long as you do not select to compare what is currently in the HTTP request with a value that requires authentication (such as LDAP attribute), the Authorization policy can be evaluated for an unauthenticated user. The following trace is for a policy with a Current Time of Day condition. The protected resource does not require authentication, so everyone can access the resource if their request comes in between 8:00 am and 5:30 pm, local time.

```
<amLogEntry> 2009-08-03T16:30:48Z INFO NIDS Application: AM#501101050:
AMDEVICEID#esp-2FA73CE1A376FD91: PolicyID#216660PM-429P-O660-N25N-L58L08MN4N5M:
NXPESID#4515: Evaluating policy </amLogEntry>

<amLogEntry> 2009-08-03T16:30:48Z INFO NIDS Application: AM#501102050:
AMDEVICEID#esp-2FA73CE1A376FD91: PolicyID#216660PM-429P-O660-N25N-L58L08MN4N5M:
NXPESID#4515: AGAuthorization Policy Trace:
   ~~RL~1~~~~Rule Count: 2~~Success(0)
   ~~RU~RuleID_1186082720202~time_of_day~DNF~~1:1~~Success(0)
   ~~CS~1~~ANDs~~1~~True(69)
   ~~CO~0~TimeOfDay(1005):::Fri Aug 03 10:30:48 MDT
2007(9:30):~com.novell.nxpe.condition.NxpeOperator@time-in-
range~(0)::::~~~True(69)
   ~~PA~1~~Permit Access~~~~Success(0)
   ~~PC~1~~Document=(ou=xpemlPEP,ou=mastercdn,ou=ContentPublisherCon
tainer,ou=Partition,ou=PartitionsContainer,ou=VCDN_Root,ou=accessManagerContainer,
o=novell:romaContentCollectionXMLDoc),Policy=(time_of_day),Rule=(1::RuleID_1186082
720202),Action=(Permit::1)~~~~Success(0)
 </amLogEntry>

<amLogEntry> 2009-08-03T16:30:48Z INFO NIDS Application: AM#501101021:
AMDEVICEID#esp-2FA73CE1A376FD91: PolicyID#216660PM-429P-O660-N25N-L58L08MN4N5M:
NXPESID#4515: Response sent: Status - success </amLogEntry>
```

The first log entry is the request to evaluate the policy. The second log entry is the evaluation of the policy, and from it you can tell that the user is not authenticated because the `AMAUTHID#` tag is missing. The third log entry is the response that is returned, and it indicates that a success was returned. The user is allowed access to the resource.

Form Fill Traces

The following sections describe how to enable logging for the Form Fill policies, describe the form that was used to create the Form Fill trace, then describe the entries that can be found in the logs:

- “Enabling Form Fill Logging” on page 1041
- “Sample Form and Policy Used for the Trace” on page 1042
- “Embedded Service Provider Trace” on page 1044
- “Proxy Service Trace” on page 1045

Enabling Form Fill Logging

Two modules evaluate the Form Fill policy and log entries:

- Embedded Service Provider (ESP) of the Access Gateway evaluates the Form Fill policy and logs entries to its file. ESP sends the messages to the `catalina.out` fileof the Access Gateway. To enable ESP logging, see “Turning on Logging for Policy Evaluation” on page 871.

- The proxy service of the Access Gateway reports on the process of finding the form data and filling it in.

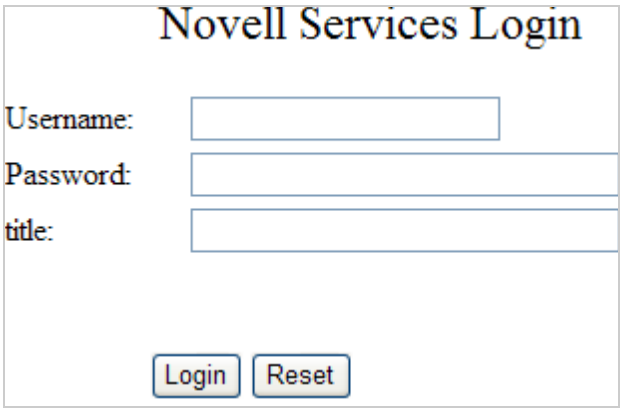
  For the Access Gateway Appliance, see the `/var/log/novell-apache2/soapmessages` file.

  You can configure a custom filter and file to log Form Fill entries. For the filter, enable the **Form Fill Processing** events in the **Advanced Log Level Options** section.

Sample Form and Policy Used for the Trace

Figure 26-10 illustrates the simple form that was used for the trace.

Figure 26-10 *Form Used for the Trace*



Source HTML for the Form

The name of the form and the fields that need to be filled in by the policy are in bold typeface.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
<head>
  <meta http-equiv="Content-type" content="text/html; charset=utf-8">
  <title>kelly</title>
</head>
<body>
  <form name="mylogin" action="double.php" method="post" id="mylogin">
    <center>
      <table border="0" cellpadding="4" cellspacing="4" width="570">
        <tr>
          <td width="121" height="285" align="left" valign="top">
          </td>
          <td width="449" height="285" align="center" valign="top">
            <p align="center">
              <font size="5">Novell Services Login<br></font>
            </p>
            <table border="0" width="86%">
              <tr>
                <td width="25%">Username:</td>
                <td width="75%">
                  <input type="TEXT" name="username">
                </td>
              </tr>
              <tr>
                <td width="25%">Password:</td>
                <td width="75%">
                  <input type="PASSWORD" name="password" size="30">
```

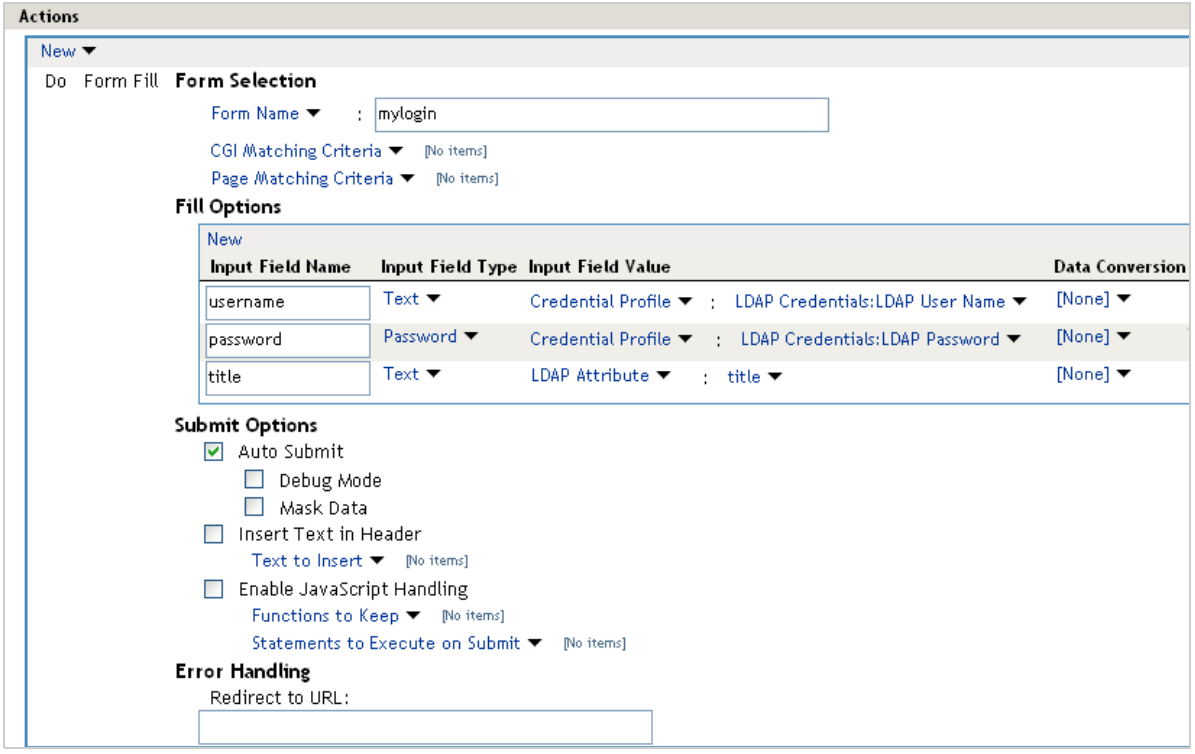
```
            </td>
          </tr>
          <tr>
            <td width="25%">title:</td>
            <td width="75%">
              <input type="TEXT" name="title" size="30">
            </td>
          </tr>
        </table>
      </td>
    </tr>

     <tr>
        <td colspan="2" align="center">
         <input type="hidden" name="formNum" value="1">
         <input type="submit" value="Login">
         <input type="reset">
        </td>
      </tr>
    </table>
  </center>
</form>
</body>
</html>
```

Form Fill Policy

The following Form Fill policy was created for the `mylogin` form. The policy is called `Form_Fill`. You can use the name of the policy to find entries for it in the log files. The policy was assigned to the `/identity/forms/simple.html` protected resource. Because the URL path identifies a specific file on the Web server, the policy does not require any CGI or page matching criteria.

Figure 26-11 *The Form Fill Policy for the mylogin Form*



This policy is configured so that the user never sees the form. Even on first login, the form is filled in for authenticated users because the user's authentication credentials are used for the username and password fields, and the title field value is obtained from the LDAP user store. If the user does not have a value for the title attribute, the user sees the form every time the page is accessed. If you want the value to be saved for these users, you need to change the policy to use a secret store rather than an LDAP attribute.

Embedded Service Provider Trace

When you look for entries for the Form_Fill policy in the Embedded Service Provider trace, you can use the following strings to find the entries:

- The name of the Form Fill policy: `Form_Fill`
- The string identifying a Form Fill trace: `AGFormFill Policy Trace`
- The policy ID (after you have found it): `PolicyID#06OO287L-06LO-KKP4-207M-6971PPM6147L`

The following trace is from the `catalina.out` file of the Embedded Service Provider of an Access Gateway Appliance. The entries have been numbered so that they can be described, and a few extra line breaks and spaces have been added to make the entries easier to read.

```
1. <amLogEntry> 2009-09-14T00:15:52Z INFO NIDS Application: AM#501101050:
AMDEVICEID#esp-917A1174C8A270FC: PolicyID#06OO287L-06LO-KKP4-207M-6971PPM6147L:
NXPESID#2663: Evaluating policy </amLogEntry>

2. <amLogEntry> 2009-09-14T00:15:52Z INFO NIDS Application: AM#501104050:
AMDEVICEID#esp-917A1174C8A270FC: PolicyID#06OO287L-06LO-KKP4-207M-6971PPM6147L:
NXPESID#2663: AGFormFill Policy Trace:
   ~~RL~1~~~~Rule Count: 1~~Success(67)
   ~~RU~RuleID_1189711482510~Form_Fill~DNF~~0:1~~Success(67)
   ~~PA~ActionID_1189711485006~~Added Form Selection Group~~~~Success
       (0)
   ~~PA~ActionID_1189711485006~~Added Fill Options Group~~~~Success(0)
   ~~PA~ActionID_1189711485006~~Added Submit Options Group~~~~Success
       (0)
   ~~PC~ActionID_1189711485006~~Document=(ou=xpemlPEP,ou=mastercdn,
     ou=ContentPublisherContainer,ou=Partition,ou=PartitionsContainer,
     ou=VCDN_Root,ou=accessManagerContainer,o=novell:romaContent
     CollectionXMLDoc),Policy=(Form_Fill),Rule=(1::RuleID_11897114
     82510),Action=(FormFill::ActionID_1189711485006)~~~~Success(0)
 </amLogEntry>

3. <amLogEntry> 2009-09-14T00:15:52Z INFO NIDS Application: AM#501101021:
AMDEVICEID#esp-917A1174C8A270FC: PolicyID#06OO287L-06LO-KKP4-207M-6971PPM6147L:
NXPESID#2663: Response sent: Status - success </amLogEntry>
```

1. The first log entry is the request to evaluate the policy. If this entry does not' occur, ensure that the Form Fill policy is enabled for the protected resource.
2. The second entry is the actual policy trace. For a Form Fill policy, it is fairly basic information about the three types of actions in the policy: matching the form, filling in the field options, and adding the submit options. To determine what information was put in the options, you need to view the proxy service trace.
3. The third entry indicates the type of response that is returned from the evaluation. In this entry, success is returned.

Proxy Service Trace

When you look for entries in the proxy trace of the Access Gateway log, you can use the following strings to find the entries:

- The name of the Form Fill policy: `Form_Fill`
- The name of the form: `mylogin`
- The names of the fill option fields: `username, password, title`

The sample trace is from the `error_log` file of a Access Gateway Appliance. Some of the lines are very long, and extra white space has been added to make them easier to read.

```
Sep  9 17:05:08 nam40-mag1 httpd[16354]: [warn] AMEVENTID#40: FF:fillSilent:
mastercdnForm_Fill3310
Sep  9 17:05:08 nam40-mag1 httpd[16354]: [warn] AMEVENTID#40: FF:Filling: username
Sep  9 17:05:08 nam40-mag1 httpd[16354]: [warn] AMEVENTID#40: FF:Filling: password
Sep  9 17:05:08 nam40-mag1 httpd[16354]: [warn] AMEVENTID#40: FF:Filling: title
Sep  9 17:05:08 nam40-mag1 httpd[16354]: [warn] AMEVENTID#40: FF: No Match
<formNum>
Sep  9 17:05:08 nam40-mag1 httpd[16354]: [warn] AMEVENTID#40: FF:fillInteractive
FormFill Policy :mastercdnForm_Fill3310 Inject JavaScript Policy:
mastercdnForm_Fill3510
Sep  9 17:05:08 nam40-mag1 httpd[16354]: [warn] AMEVENTID#42: FF:fillSilent:
mastercdnForm_Fill3310, referer: http://www.ag1.com/identity/forms/simple.html
Sep  9 17:05:08 nam40-mag1 httpd[16354]: [warn] AMEVENTID#42: FF:Not Found: <form>,
referer: http://www.ag1.com/identity/forms/simple.html
Sep  9 17:05:08 nam40-mag1 httpd[16354]: [warn] AMEVENTID#42: FF:no <Form
pol:mastercdnForm_Fill3310, referer: http://www.ag1.com/identity/forms/simple.html
```

On the Access Gateway Appliance, you can get more detailed information about the process that was used to fill the form when you turn on logging to the `soapmessages` file.

Access Manager Audit Events and Data

This section contains all the audit events logged by Access Manager Appliance. Each event has EventID, Description, Originator Title, Target Title, Subtarget Title, Text1 Title, Text2 Title, Text3 Title, Value1 Title, Value1 Type, Group Title, Data Length, and Data Type values stored. Each field contains a single character token (such as B, U, Y, and so on) that represent the data fields of the audit event, with each letter representing a different data field. The mapping of the character tokens to data fields is found in the `nids_en.lsc`file.

Access Manager is listed among the log applications on the **General** tab on the Logging Server Options page (**Auditing and Logging > Logging Server Options**). You can view events on the Event list page in **Auditing and Logging** > **Logging Server Options** > **[Name of Novell Audit Secure Logging Server]** > **Novell Access Manager** > **Events**.

When you run an SQL query (**Auditing and Logging > Queries > [Name] > Run**), the system displays the results on the Query Results page. The **EventID** column displays the description of the event. Below, the event ID is listed with the description, to help you quickly locate the data for each audit event.

JavaScript Object Notation (JSON) Event Format

Sample JSON Format

This event is generated when you select the Risk-Based Authentication Succeeded option under Novell Audit Logging on the Logging page of an Identity Server configuration.

The following is a sample JSON event format of a Risk-Based authentication:

```
{
"appName" : "Novell Access Manager",
"Component" : "nidp",
"timeStamp" : "Fri, 31 Jul 2015 17:30:57 +0530",
"eventId" : "002E0025",
"Description": "NIDS: Risk based additional authentication executed successfully
for user",
"Originator": "9772686A5705BA6C",
"Target": "cn=admin,o=novell",
"SubTarget": "3883A05A302BA3BDC7899AF05810B08B",
"stringValue1": "35",
"stringValue2": "medium",
"stringValue3": "null",
"numericValue1": "0",
"numericValue2": "0",
"numericValue3": "0",
"Data": "MTY0Ljk5LjEzNy41Mg==",
"Message": "[Fri, 31 Jul 2015 17:30:57 +0530] [Novell Access Manager\nidp]:
AMDEVICEID#9772686A5705BA6C: AMAUTHID#3883A05A302BA3BDC7899AF05810B08B: Risk based
authentication successful for user: [cn=admin,o=novell]. RiskScore: [35] RiskLevel:
[Medium] Additional authentication class: [$SF] Client IP Address:
[164.99.137.52]",
}
```

The following table lists the event fields with its corresponding description:

Field	Description
appName	Specifies the name of the product.
Component	Specifies the name of the NetIQ Access Manager component. For example, “nipd” identifies that the audit is triggered by the Identity Server.
timeStamp	Specifies the time when the event occurred.
eventId	Specifies the event ID. For example, 002E0025. To view all the events and their corresponding event IDs, see the below sections.
Description	Describes the event.
Originator	Specifies the ID of the device that generated this event. For example, 9772686A5705BA6C is the device with ID “idp-9772686A5705BA6C”
Target	Specifies the target on which this action is executed. In the above example, the action is risk-based authentication, hence the target is the user id for which the risk was assessed.
SubTarget	Specifies the additional details about the target.
stringValue1	Specifies an event-specific string value. The value of this field varies from event to event. For example, it is null if the event has no value to pass.
stringValue2	Specifies an event-specific string value. The value of this field varies from event to event. For example, it is null if the event has no value to pass.

Field	Description
stringValue3	Specifies an event-specific string value. The value of this field varies from event to event. For example, it is null if the event has no value to pass.
numbericValue1	Specifies an event-specific string value. The value of this field varies from event to event. For example, it is null if the event has no value to pass.
numbericValue2	Specifies an event-specific string value. The value of this field varies from event to event. For example, it is null if the event has no value to pass.
numbericValue3	Specifies an event-specific string value. The value of this field varies from event to event. For example, it is null if the event has no value to pass.
Data	Specifies an event-specific data.
Message	Specifies a friendly detailed message related to the event.

NOTE: The Syslog agents use the following message format: `rfc3164`. For more information, see RFC 3164 documentation.

This section discusses the following audit events:

- "NIDS: Sent a Federate Request (002e0001)" on page 1049
- "NIDS: Received a Federate Request (002e0002)" on page 1049
- "NIDS: Sent a Defederate Request (002e0003)" on page 1050
- "NIDS: Received a Defederate Request (002e0004)" on page 1050
- "NIDS: Sent a Register Name Request (002e0005)" on page 1051
- "NIDS: Received a Register Name Request (002e0006)" on page 1051
- "NIDS: Logged Out an Authentication that Was Provided to a Remote Consumer (002e0007)" on page 1052
- "NIDS: Logged out a Local Authentication (002e0008)" on page 1052
- "NIDS: Provided an Authentication to a Remote Consumer (002e0009)" on page 1053
- "NIDS: User Session Was Authenticated (002e000a)" on page 1053
- "NIDS: Failed to Provide an Authentication to a Remote Consumer (002e000b)" on page 1054
- "NIDS: User Session Authentication Failed (002e000c)" on page 1054
- "NIDS: Received an Attribute Query Request (002e000d)" on page 1055
- "NIDS: User Account Provisioned (002e000e)" on page 1055
- "NIDS: Failed to Provision a User Account (002e000f)" on page 1056
- "NIDS: Web Service Query (002e0010)" on page 1056
- "NIDS: Web Service Modify (002e0011)" on page 1057
- "NIDS: Connection to User Store Replica Lost (002e0012)" on page 1057
- "NIDS: Connection to User Store Replica Reestablished (002e0013)" on page 1058
- "NIDS: Server Started (002e0014)" on page 1058
- "NIDS: Server Stopped (002e0015)" on page 1059

- “NIDS: Server Refreshed (002e0016)” on page 1059
- “NIDS: Intruder Lockout (002e0017)” on page 1060
- “NIDS: Severe Component Log Entry (002e0018)” on page 1060
- “NIDS: Warning Component Log Entry (002e0019)” on page 1061
- “NIDS: Failed to Broker an Authentication from Identity Provider to Service Provider as Identity Provider and Service Provider Are not in Same Group (002E001A)” on page 1061
- “NIDS: Failed to Broker an Authentication from Identity Provider to Service Provider Because a Policy Evaluated to Deny (002E001B)” on page 1062
- “NIDS: Brokered an Authentication from Identity Provider to Service Provider (002E001C)” on page 1062
- “NIDS: OAuth2 Authorization code issued (002e0028)” on page 1063
- “NIDS: OAuth2 token issued (002e0029)” on page 1063
- “NIDS: OAuth2 Authorization code issue failed (002e0030)” on page 1064
- “NIDS: OpenID token issued (002e0031)” on page 1064
- “NIDS: OAuth2 refresh token issued (002e0032)” on page 1065
- “NIDS: OAuth2 token issue failed (002e0033)” on page 1065
- “NIDS: OpenID token issue failed (002e0034)” on page 1066
- “NIDS: OAuth2 refresh token issue failed (002e0035)” on page 1066
- “NIDS: OAuth2 client has been registered successfully (002e0036)” on page 1066
- “NIDS: OAuth2 client has been modified successfully (002e0037)” on page 1067
- “NIDS: OAuth2 client has been deleted successfully (002e0038)” on page 1067
- “NIDS: OAuth2 user has provided consent (002e0039)” on page 1068
- “NIDS: OAuth2 user has revoked consent (002e0040)” on page 1068
- “NIDS: OAuth2 token validation success (002e0041)” on page 1069
- “NIDS: OAuth2 token validation failed (002e0042)” on page 1069
- “NIDS: OAuth2 client registration failed (002e0043)” on page 1070
- “NIDS: Roles PEP Configured (002e0300)” on page 1070
- “Access Gateway: PEP Configured (002e0301)” on page 1070
- “Roles Assignment Policy Evaluation (002e0320)” on page 1071
- “Access Gateway: Authorization Policy Evaluation (002e0321)” on page 1071
- “Access Gateway: Form Fill Policy Evaluation (002e0322)” on page 1072
- “Access Gateway: Identity Injection Policy Evaluation (002e0323)” on page 1072
- “Access Gateway: Access Denied (0x002e0505)” on page 1073
- “Access Gateway: URL Not Found (0x002e0508)” on page 1073
- “Access Gateway: System Started (0x002e0509)” on page 1074
- “Access Gateway: System Shutdown (0x002e050a)” on page 1075
- “Access Gateway: Identity Injection Parameters (0x002e050c)” on page 1075
- “Access Gateway: Identity Injection Failed (0x002e050d)” on page 1076
- “Access Gateway: Form Fill Authentication (0x002e050e)” on page 1076
- “Access Gateway: Form Fill Authentication Failed (0x002e050f)” on page 1077
- “Access Gateway: URL Accessed (0x002e0512)” on page 1078

- “Access Gateway: IP Access Attempted (0x002e0513)” on page 1078
- “Access Gateway: Webserver Down (0x002e0515)” on page 1079
- “Access Gateway: All WebServers for a Service is Down (0x002e0516)” on page 1079
- “Management Communication Channel: Health Change (0x002e0601)” on page 1080
- “Management Communication Channel: Device Imported (0x002e0602)” on page 1081
- “Management Communication Channel: Device Deleted (0x002e0603)” on page 1081
- “Management Communication Channel: Device Configuration Changed (0x002e0604)” on page 1082
- “Management Communication Channel: Device Alert (0x002e0605)” on page 1082
- “Risk-Based Authentication: 002e0025” on page 1083
- “Risk-Based Authentication: 002e0026” on page 1083
- “Risk-Based Authentication: 002e0027” on page 1084

NIDS: Sent a Federate Request (002e0001)

This event is generated when you select the **Federation Request Sent** option under **Novell Audit Logging** on the Logging page of an Identity Server configuration.

Description: NIDS: Sent a federate request.

Originator (B): Schema Title: Originator Data Description: JCC Device ID (AMDEVICEID#device_id:)

Target (U): Schema Title: User Identifier Data Description: LDAP Auth: User DN Other Auth: User GUID

SubTarget (Y): null

Text1 (S): null

Text2 (T): null

Text3 (F): null

Value1 (1): 0

Group (G): 0

Data Length (X): 0

Data (D): null

NIDS: Received a Federate Request (002e0002)

This event is generated when you select the **Federation Request Handled** option under **Novell Audit Logging** on the Logging page of an Identity Server configuration.

Description: NIDS: Received a federate request.

Originator (B): Schema Title: Originator Data Description: JCC Device ID (AMDEVICEID#device_id:)

Target (U): Schema Title: User Identifier Data Description: LDAP Auth: User DN Other Auth: User GUID

SubTarget (Y): null

Text1 (S): Schema Title: Provider Identifier; Data Description: Service Provider ID

Text2 (T): null

Value1 (1): 0

Group (G): 0

Data Length (X): 0

Data (D): null

NIDS: Sent a Defederate Request (002e0003)

This event is generated when you select the **Defederation Request Sent** option under **Novell Audit Logging** on the Logging page of an Identity Server configuration.

Description: NIDS: Sent a defederate request.

Originator (B): Schema Title: Originator Data Description: JCC Device ID (AMDEVICEID#device_id:)

Target (U): Schema Title: User Identifier Data Description: LDAP Auth: User DN Other Auth: User GUID

SubTarget (Y): null

Text1 (S): Schema Title: Provider Identifier; Data Description: Service Provider ID

Text2 (T): null

Value1 (1): 0

Group (G): 0

Data Length (X): 0

Data (D): null

NIDS: Received a Defederate Request (002e0004)

This event is generated when you select the **Defederation Request Handled** option under **Novell Audit Logging** on the Logging page of an Identity Server configuration.

Description: NIDS: Received a defederate request

Originator (B): Schema Title: Originator Data Description: JCC Device ID (AMDEVICEID#device_id:)

Target (U): Schema Title: User Identifier Data Description: LDAP Auth: User DN Other Auth: User GUID

SubTarget (Y): null

Text1 (S): Schema Title: Provider Identifier Data Description: Service Provider ID

Text2 (T): null

Text3 (F): null

Value1 (1): 0

Group (G): 0

Data Length (X): 0

Data (D): null

NIDS: Sent a Register Name Request (002e0005)

Description: NIDS: Sent a register name request

Originator (B): Schema Title: Originator Data Description: JCC Device ID (AMDEVICEID#device_id:)

Target (U): null

SubTarget (Y): null

Text1 (S): null

Text2 (T): null

Text3 (F): null

Value1 (1): 0

Group (G): 0

Data Length (X): 0

Data (D): null

NIDS: Received a Register Name Request (002e0006)

This event is generated when you select the **Register Name Request Handled** option under **Novell Audit Logging** on the Logging page of an Identity Server configuration.

Description: NIDS: Received a register name request

Originator (B): Schema Title: Originator Data Description: JCC Device ID (AMDEVICEID#device_id:)

Target (U): null

SubTarget (Y): null

Text1 (S): null

Text2 (T): null

Text3 (F): null

Value1 (1): 0

Group (G): 0

Data Length (X): 0

Data (D): null

NIDS: Logged Out an Authentication that Was Provided to a Remote Consumer (002e0007)

This event is generated when you select the **Logout Provided** option under **Novell Audit Logging** on the Logging page of an Identity Server configuration.

Description: NIDS: Logged out an authentication that was provided to a remote consumer

Originator (B): Schema Title: Originator Data Description: JCC Device ID (AMDEVICEID#device_id:)

Target (U): Schema Title: User Identifier Data Description: LDAP Auth: User DN Other Auth: User GUID

SubTarget (Y): null

Text1 (S): Schema Title: Authentication Identifier Data Description: IDP Session ID (AMAUTHID#auth_id:)

Text2 (T): null

Text3 (F): null

Value1 (1): Schema Title: Timed Out Data Description: 0 = other reason 1 = timed out

Group (G): 0

Data Length (X): 0

Data (D): null

NIDS: Logged out a Local Authentication (002e0008)

This event is generated when you select the **Logout Local** option under **Novell Audit Logging** on the Logging page of an Identity Server configuration.

Description: NIDS: Logged out a local authentication

Originator (B): Schema Title: Originator Data Description: JCC Device ID (AMDEVICEID#device_id:)

Target (U): Schema Title: User Identifier Data Description: LDAP Auth: User DN Other Auth: User GUID

SubTarget (Y): null

Text1 (S): Schema Title: Authentication Identifier Data Description: IDP Session ID (AMAUTHID#auth_id:

Text2 (T): null

Text3 (F): null

Value1 (1): Schema Title: Timed Out Data Description: 0 = other reason 1 = timed out

Group (G): 0

Data Length (X): 0

Data (D): null

NIDS: Provided an Authentication to a Remote Consumer (002e0009)

This event is generated when you select the **Login Consumed** option under **Novell Audit Logging** on the Logging page of an Identity Server configuration.

Description: NIDS: Provided an authentication to a remote consumer

Originator (B): Schema Title: Originator Data Description: JCC Device ID (AMDEVICEID#device_id:)

Target (U): Schema Title: User Identifier Data Description: User DN

SubTarget (Y): Schema Title: Authentication Identifier Data Description: IDP Session ID (AMAUTHID#auth_id:)

Text1 (S): Schema Title: Authentication Type Data Description: Authentication Profile

Text2 (T): Schema Title: Authentication Entity Name Data Description: Authentication Source

Text3 (F): Schema Title: Contract Class or Method Name Data Description: Authentication Contract URI

Value1 (1): 0

Group (G): 0

Data Length (X): 0

Data (D): Schema Title: Client IP Address Description: IP Address of the host from which the authentication succeeded.

NIDS: User Session Was Authenticated (002e000a)

This event is generated when you select the **Login Provided** option under **Novell Audit Logging** on the Logging page of an Identity Server configuration.

Description: NIDS: User session was authenticated

Originator (B): Schema Title: Originator Data Description: JCC Device ID (AMDEVICEID#device_id:)

Target (U): Schema Title: User Identifier Data Description: User DN

SubTarget (Y): Schema Title: Authentication Identifier Data Description: IDP Session ID (AMAUTHID#auth_id:)

Text1 (S): Schema Title: Authentication Type Data Description: Authentication Profile

Text2 (T): Schema Title: Authentication Entity Name Data Description: Authentication Source

Text3 (F): Schema Title: Contract Class or Method Name Data Description: Authentication Contract URI

Value1 (1): 0

Group (G): 0

Data Length (X): 0

Data (D): Schema Title: Client IP Address Description: IP Address of the host from which the authentication succeeded.

NIDS: Failed to Provide an Authentication to a Remote Consumer (002e000b)

This event is generated when you select the **Login Consumed Failure** option under **Novell Audit Logging** on the Logging page of an Identity Server configuration.

Description: NIDS: Failed to provide an authentication to a remote consumer

Originator (B): Schema Title: Originator Data Description: JCC Device ID (AMDEVICEID#device_id:)

Target (U): Schema Title: User Identifier Data Description: User DN

SubTarget (Y): null

Text1 (S): Schema Title: Authentication Identifier Data Description: IDP Session ID (AMAUTHID#auth_id:)

Text2 (T): Schema Title: Provider Identifier Data Description: Service Provider ID

Text3 (F): Schema Title: Reason Data Description: Reason Message

Value1 (1): 0

Group (G): 0

Data Length (X): 0

Data (D): null

NIDS: User Session Authentication Failed (002e000c)

This event is generated when you select the **Login Provided Failure** option under **Novell Audit Logging** on the Logging page of an Identity Server configuration. Use the **Description** field and the **Text3 (F)** field to determine whether the failure came from a contract, SAML 1.1, SAML 2.0, or Liberty.

Description: `NIDS: User session authentication failed`. This string plus one of the following phrases: for a contract failure, `Contract Execution`; for a SAML 1.1 failure, `SAML Assertion`; for a SAML 2.0 failure, `SAML2 SSO`; for a Liberty failure, `Liberty SSO`.

Originator (B): Schema Title: Originator Data Description: JCC Device ID (AMDEVICEID#device_id:)

Target (U): Schema Title: Authentication Contract Name Data Description: Contract URI

SubTarget (Y): Schema Title: User Identifier Data Description: User DN

Text1 (S): Schema Title: Authentication Identifier Data Description: IDP Session ID (AMAUTHID#auth_id:)

Text2 (T): Schema Title: Reason Data Description: Reason Message

Text3 (F): Schema Title: Authentication Source Data Description: For a contract, contains the authentication method name; for Liberty, contains the service provider IP; for SAML 1.1, contains the SAML assertion issuer; for SAML 2.0, contains the service provider IP.

Value1 (1): 0

Group (G): 0

Data Length (X): 0

Data (D): Schema Title: Client IP Address Description: IP Address of the host from which the authentication failed.

NIDS: Received an Attribute Query Request (002e000d)

This event is generated when you select the **Attribute Query Request Handled** option under **Novell Audit Logging** on the Logging page of an Identity Server configuration.

Description: NIDS: Received an attribute query request

Originator (B): Schema Title: Originator Data Description: JCC Device ID (AMDEVICEID#device_id:)

Target (U): Schema Title: User Identifier Data Description: LDAP Auth: User DN Other Auth: User GUID

SubTarget (Y): null

Text1 (S): Schema Title: Provider Identifier Data Description: Service Provider ID

Text2 (T): Schema Title: Attribute Names Data Description: Requested Attributes

Text3 (F): null

Value1 (1): 0

Group (G): 0

Data Length (X): 0

Data (D): null

NIDS: User Account Provisioned (002e000e)

This event is generated when you select the **User Account Provisioned** option under **Novell Audit Logging** on the Logging page of an Identity Server configuration.

Description: NIDS: User account provisioned

Originator (B): Schema Title: Originator Data Description: JCC Device ID (AMDEVICEID#device_id:)

Target (U): Schema Title: User Store Identifier Data Description: Displayable user name

SubTarget (Y): null

Text1 (S): Schema Title: User Identifier Data Description: Authentication User Name

Text2 (T): Schema Title: Authentication Identifier Data Description: IDP Session ID (AMAUTHID#auth_id:)

Text3 (F): null

Value1 (1): 0

Group (G): 0

Data Length (X): 0

Data (D): null

NIDS: Failed to Provision a User Account (002e000f)

This event is generated when you select the **User Account Provisioned Failure** option under **Novell Audit Logging** on the Logging page of an Identity Server configuration.

Description: NIDS: Failed to provision a user account

Originator (B): Schema Title: Originator Data Description: JCC Device ID (AMDEVICEID#device_id:)

Target (U): Schema Title: User Store Identifier Data Description: Displayable User Name

SubTarget (Y): null

Text1 (S): Schema Title: User Identifier Data Description: Authentication User Name

Text2 (T): Schema Title: Reason Data Description: Reason Message

Text3 (F): null

Value1 (1): 0

Group (G): 0

Data Length (X): 0

Data (D): null

NIDS: Web Service Query (002e0010)

This event is generated when you select the **Web Service Query Handled** option under **Novell Audit Logging** on the Logging page of an Identity Server configuration. The Identity Server uses this event for two types of Web service queries:

- **Discovery:** This is a query to discover a service. For this type of query, the **Group (G)** field is not used. For a remote query, the **Data Description** of the **Value1** field is set to 0. For a local query, the **Data Description** of the **Value1** field is set to 1.
- **Profile:** This is a query to get attributes for a user from a profile (personal, credential, etc.). For this type of query, the **Group (G)** field contains a GroupingID for all attributes selected in the request. A separate event is generated for each attribute select list in the request. For a remote query, the **Data Description** of the **Value1** field is set to 0. For a local query, the **Data Description** of the **Value1** field is set to 1.

Description: NIDS: Web Service query

Originator (B): Schema Title: Originator Data Description: JCC Device ID (AMDEVICEID#device_id:)

Target (U): Schema Title: User Identifier Data Description: User DN

SubTarget (Y): null

Text1 (S): Schema Title: Provider Identifier Data Description: Requesting Provider ID

Text2 (T): Schema Title: Select String Data Description: Requested attributes; select string

Text3 (F): Schema Title: Service Identifier Data Description: Web Service URI

Value1 (1): Schema Title: Local Data Description: 0 – Remote 1 – Local

Group (G): Schema Title: Query Group Data Description: If this is a profile query, it contains the grouping ID for all attributes selected in this request. Otherwise, this field is not used in the event.

Data Length (X): 0

Data (D): null

NIDS: Web Service Modify (002e0011)

This event is generated when you select the **Web Service Modify Handled** option under **Novell Audit Logging** on the Logging page of an Identity Server configuration. The Identity Server uses this event for two types of Web service modify requests:

- **Discovery:** This is a request to discover a service to modify. For this type of request, the **Group (G)** field is not used. For a remote request, the **Data Description** of the **Value1** field is set to 0. For a local request, the **Data Description** of the **Value1** field is set to 1.
- **Profile:** This is a request to modify the attributes of a user in a profile (personal, credential, etc.). For this type of request, the **Group (G)** field contains a GroupingID for all attributes selected in the request. A separate event is generated for each attribute select list in the modify request. For a remote request, the **Data Description** of the **Value1** field is set to 0. For a local request, the **Data Description** of the **Value1** field is set to 1.

Description: NIDS: Web Service modify

Originator (B): Schema Title: Originator Data Description: JCC Device ID (AMDEVICEID#device_id:)

Target (U): Schema Title: User Identifier Data Description: User DN

SubTarget (Y): null

Text1 (S): Schema Title: Provider Identifier Data Description: Requesting Provider ID

Text2 (T): Schema Title: Select String Data Description: Modified attributes select string

Text3 (F): Schema Title: Service Identifier Data Description: Web Service URI

Value1 (1): Schema Title: Local Data Description: 0 – Remote; 1 – Local

Group (G): Schema Title: Modify Group Data Description: If this is a profile modify, it contains the grouping ID for each attribute select list in the request. Otherwise, this field is not used in the event.

Data Length (X): 0

Data (D): null

NIDS: Connection to User Store Replica Lost (002e0012)

This event is generated when you select the **LDAP Connection Lost** option under **Novell Audit Logging** on the Logging page of an Identity Server configuration.

Description: NIDS: Connection to user store replica lost

Originator (B): Schema Title: Originator Data Description: JCC Device ID (AMDEVICEID#device_id:)

Target (U): Schema Title: User Store Replica Name Data Description: Replica name

SubTarget (Y): null

Text1 (S): Schema Title: User Store Replica Host Data Description: IP Address of User Store replica server

Text2 (T): null

Text3 (F): null

Value1 (1): 0

Group (G): 0

Data Length (X): 0

Data (D): null

NIDS: Connection to User Store Replica Reestablished (002e0013)

This event is generated when you select the **LDAP Connection Reestablished** option under **Novell Audit Logging** on the Logging page of an Identity Server configuration.

Description: NIDS: Connection to user store replica reestablished

Originator (B): Schema Title: Originator Data Description: JCC Device ID (AMDEVICEID#device_id:)

Target (U): Schema Title: User Store Replica Name Data Description: Replica name

SubTarget (Y): null

Text1 (S): Schema Title: User Store Replica Host Data Description: IP Address of User Store replica server

Text2 (T): null

Text3 (F): null

Value1 (1): 0

Group (G): 0

Data Length (X): 0

Data (D): null

NIDS: Server Started (002e0014)

This event is generated when you select the **Server Started** option under **Novell Audit Logging** on the Logging page of an Identity Server configuration.

Description: NIDS: Server started

Originator (B): Schema Title: Originator Data Description: JCC Device ID (AMDEVICEID#device_id:)

Target (U): Schema Title: Configuration Identifier Data Description: Configuration Object DN

SubTarget (Y): null

Text1 (S): Schema Title: Server Identifier Data Description: Unique server ID also used to create Liberty and SAML artifacts

Text2 (T): null

Text3 (F): null

Value1 (1): 0

Group (G): 0

Data Length (X): 0

Data (D): null

NIDS: Server Stopped (002e0015)

This event is generated when you select the **Server Stopped** option under **Novell Audit Logging** on the Logging page of an Identity Server configuration.

Description: NIDS: Server stopped

Originator (B): Schema Title: Originator Data Description: JCC Device ID (AMDEVICEID#device_id:)

Target (U): Schema Title: Configuration Identifier Data Description: Configuration object DN

SubTarget (Y): null

Text1 (S): Schema Title: Server Identifier Data Description: Unique server ID also used to create Liberty and SAML artifacts

Text2 (T): null

Text3 (F): null

Value1 (1): 0

Group (G): 0

Data Length (X): 0

Data (D): null

NIDS: Server Refreshed (002e0016)

This event is generated when you select the **Server Refreshed** option under **Novell Audit Logging** on the Logging page of an Identity Server configuration.

Description: NIDS: Server Refreshed

Originator (B): Schema Title: Originator Data Description: JCC Device ID (AMDEVICEID#device_id:)

Target (U): Schema Title: Configuration Identifier Data Description: Configuration Object DN

SubTarget (Y): null

Text1 (S): Schema Title: Server Identifier Data Description: Unique server ID also used to create Liberty and SAML artifacts

Text2 (T): null

Text3 (F): null

Value1 (1): 0

Group (G): 0

Data Length (X): 0

Data (D): null

NIDS: Intruder Lockout (002e0017)

This event is generated when you select the **Intruder Lockout Detected** option under **Novell Audit Logging** on the Logging page of an Identity Server configuration.

Description: NIDS: Intruder Lockout

Originator (B): Schema Title: Originator Data Description: JCC Device ID (AMDEVICEID#device_id:)

Target (U): Schema Title: User Identifier Data Description: User DN

SubTarget (Y): null

Text1 (S): Schema Title: Server Identifier Data Description: IP address of the User Store replica server

Text2 (T): null

Text3 (F): null

Value1 (1): 0

Group (G): 0

Data Length (X): 0

Data (D): null

NIDS: Severe Component Log Entry (002e0018)

This event is generated when you select the **Component Log Severe Messages** option under **Novell Audit Logging** on the Logging page of an Identity Server configuration.

Description: NIDS: Severe Component Log Entry

Originator (B): Schema Title: Originator Data Description: JCC Device ID (AMDEVICEID#device_id:)

Target (U): null

SubTarget (Y): null

Text1 (S): Schema Title: Component Log Text Data Description: Server Error Text

Text2 (T): null

Text3 (F): null

Value1 (1): 0

Group (G): 0

Data Length (X): 0

Data (D): null

NIDS: Warning Component Log Entry (002e0019)

This event is generated when you select the **Component Log Warning Messages** option under **Novell Audit Logging** on the Logging page of an Identity Server configuration.

Description: NIDS: Warning Component Log Entry

Originator (B): Schema Title: Originator Data Description: JCC Device ID (AMDEVICEID#device_id:)

Target (U): null

SubTarget (Y): null

Text1 (S): Schema Title: Component Log Text Data Description: Warning Error Text

Text2 (T): null

Text3 (F): null

Value1 (1): 0

Group (G): 0

Data Length (X): 0

Data (D): null

NIDS: Failed to Broker an Authentication from Identity Provider to Service Provider as Identity Provider and Service Provider Are not in Same Group (002E001A)

This event is generated when you select the **Brokering Across Groups Denied** option under **Novell Audit Logging** on the Logging page of an Identity Server configuration.

Description: NIDS: Failed to broker an authentication from identity provider to service provider as identity provider and service provider are not in same group

Originator (B): Schema Title: Originator Data Description: JCC Device ID (AMDEVICEID#device_id:)

Target (U): Schema Title: User Identifier Data Description: User DN

SubTarget (Y): null

Text1 (S): Schema Title: Identity Provider IdentifierDescription : Identity Provider ID

Text2 (T): Schema Title: Service Provider IdentifierDescription: Service Provider ID

Text3 (F): null

Value1 (1): 0

Group (G): 0

Data Length (X): Schema Title: Target URL Length Description: Byte length of the target URL

Data (D): Schema Title: Target URL Description: Target URL

NIDS: Failed to Broker an Authentication from Identity Provider to Service Provider Because a Policy Evaluated to Deny (002E001B)

This event is generated when you select the **Brokering Rule Evaluated to Deny** option under **Novell Audit Logging** on the Logging page of an Identity Server configuration.

Description: NIDS: Failed to broker an authentication from identity provider to service provider because a policy evaluated to deny

Originator (B): Schema Title: Originator Data Description: JCC Device ID (AMDEVICEID#device_id:)

Target (U): Schema Title: User Identifier Data Description: User DN

SubTarget (Y): Schema Title: Broker Group Name Description: Name of the Brokering Group

Text1 (S): Schema Title: Identity Provider IdentifierDescription: Identity Provider ID

Text2 (T): Schema Title: Service Provider IdentifierDescription: Service Provider ID

Text3 (F): Schema Title: Broker Policy Description: Name of the Broker Policy that evaluated to deny

Value1 (1): 0

Group (G): 0

Data Length (X): Schema Title: Target URL Length Description: Byte length of the target URL

Data (D): Schema Title: Target URL Description: Target URL

NIDS: Brokered an Authentication from Identity Provider to Service Provider (002E001C)

This event is generated when you select the **Brokering Handled** option under **Novell Audit Logging** on the Logging page of an Identity Server configuration.

Description: NIDS: Brokered an authentication from identity provider to service provider

Originator (B): Schema Title: Originator Data Description: JCC Device ID (AMDEVICEID#device_id:)

Target (U): Schema Title: User Identifier Data Description: User DN

SubTarget (Y): Schema Title: Broker Group Name Description: Name of the Brokering Group

Text1 (S): Schema Title: Identity Provider Identifier Description: Identity Provider ID

Text2 (T): Schema Title: Service Provider Identifier Description: Service Provider ID

Text3 (F): null

Value1 (1): 0

Group (G): 0

Data Length (X): Schema Title: Target URL Length Description: Byte length of the target URL

Data (D): Schema Title: Target URL Description: Target URL

NIDS: OAuth2 Authorization code issued (002e0028)

This event is generated when you select the **OAuth & OpenID Token Issued** option under **Novell Audit Logging** on the Logging page of an Identity Server configuration.

Description: NIDS: OAuth2 Authorization code issued

Originator (B): Schema Title: Originator

Data Description: JCC Device ID (AMDEVICEID#device_id:)

Target (U): Schema Title: User Identifier

Data Description: Authentication User Name

SubTarget (Y): null

Text1 (S): Schema Title: Issued At Data

Data Description: Token issued time stamp in Millisecond

Text2 (T): Schema Title: Issued To Data

Description: Client Name

Text3 (F): Schema Title: Validity Data

Description: From: Time in Milliseconds - To: Time in Milliseconds

NIDS: OAuth2 token issued (002e0029)

This event is generated when you select the **OAuth & OpenID Token Issued** option under **Novell Audit Logging** on the Logging page of an Identity Server configuration.

Description: NIDS: OAuth2 token issued

Originator (B): Schema Title: Originator

Data Description: JCC Device ID (AMDEVICEID#device_id:)

Target (U): Schema Title: User Identifier

Data Description: Authentication User Name

SubTarget (Y): Schema Title: Grant Type

Data Description: Oauth grant type

Text1 (S): Schema Title: Issued At

Data Description: Token issued time stamp in Milliseconds

Text2 (T): Schema Title: Issued To

Data Description: Client Name

Text3 (F): Schema Title: Validity

Data Description: From: Time in Milliseconds - To: Time in Milliseconds

NIDS: OAuth2 Authorization code issue failed (002e0030)

This event is generated when you select the **OAuth & OpenID Token Issue Failed** option under **Novell Audit Logging** on the Logging page of an Identity Server configuration.

Description: NIDS: OAuth2 Authorization code issue failed

Originator (B): Schema Title: Originator

Data Description: JCC Device ID (AMDEVICEID#device_id:)

Target (U): Schema Title: User Identifier

Data Description: Authentication User Name

SubTarget (Y): null

Text1 (S): Schema Title: Failed At

Data Description: Code issued failed time stamp in Milliseconds

Text2 (T): Schema Title: Reason

Data Description: Reason for failure

Text3 (F): null

NIDS: OpenID token issued (002e0031)

This event is generated when you select the **OAuth & OpenID Token Issue** option under **Novell Audit Logging** on the Logging page of an Identity Server configuration.

Description: NIDS: OpenID token issued

Originator (B): Schema Title: Originator

Data Description: JCC Device ID (AMDEVICEID#device_id:)

Target (U): Schema Title: User Identifier

Data Description: Authentication User Name

SubTarget (Y): null

Text1(S): Schema Title: Issued At

Data Description: ID Token issued time stamp in Millisecond

Text2(T): Schema Title: Issued To

Data Description: Client Name s

Text3(F): Schema Title: Expires

Data Description: Expires in second

NIDS: OAuth2 refresh token issued (002e0032)

This event is generated when you select the **OAuth & OpenID Token Issue** option under **Novell Audit Logging** on the Logging page of an Identity Server configuration.

Description: NIDS: OAuth2 refresh token issued

Originator (B): Schema Title: Originator

Data Description: JCC Device ID (AMDEVICEID#device_id:)

Target (U): Schema Title: User Identifier

Data Description: Authentication User Name

SubTarget (Y): null

Text1 (S): Schema Title: Issued At

Data Description: Token issued time stamp in Millisecond

Text2 (T): Schema Title: Issued To

Data Description: Client Name

Text3 (F): Schema Title: Validity

Data Description: From: Time in Milliseconds - To: Time in Milliseconds

NIDS: OAuth2 token issue failed (002e0033)

This event is generated when you select the **OAuth & OpenID Token Issue Failed** option under **Novell Audit Logging** on the Logging page of an Identity Server configuration.

Description: NIDS: OAuth2 token issue failed

Originator (B): Schema Title: Originator

Data Description: JCC Device ID (AMDEVICEID#device_id:)

Target (U): Schema Title: User Identifier

Data Description: Authentication User Name

SubTarget (Y): Schema Title: Grant Type

Data Description: Oauth grant type

Text1 (S): Schema Title: Failed At

Data Description: Token issue failed time stamp in Milliseconds

Text2 (T): Schema Title: Issued To

Data Description: Client Name

Text3 (F): Schema Title: Reason

Data Description: Reason for failure

NIDS: OpenID token issue failed (002e0034)

This event is generated when you select the **OAuth & OpenID Token Issue Failed** option under **Novell Audit Logging** on the Logging page of an Identity Server configuration.

Description: NIDS: OpenID token issue failed

Originator (B): Schema Title: Originator

Data Description: JCC Device ID (AMDEVICEID#device_id:)

Target (U): Schema Title: User Identifier Data Description: Authentication User Name

SubTarget (Y): null

Text1 (S): Schema Title: Failed At

Data Description: Token issue failed time stamp in Milliseconds

Text2 (T): Schema Title: Issued To

Data Description: Client Name

Text31 (F): Schema Title: Reason

Data Description: Reason for failure

NIDS: OAuth2 refresh token issue failed (002e0035)

This event is generated when you select the **OAuth & OpenID Token Issue Failed** option under **Novell Audit Logging** on the Logging page of an Identity Server configuration.

Description: NIDS: OAuth2 refresh token issue failed

Originator (B): Schema Title: Originator

Data Description: JCC Device ID (AMDEVICEID#device_id:)

Target (U): Schema Title: User Identifier

Data Description: Authentication User Name

SubTarget (Y): null

Text1 (S): Schema Title: Failed At

Data Description: Token issue failed time stamp in Milliseconds

Text2 (T): Schema Title: Issued To

Data Description: Client Name

Text31 (F): Schema Title: Reason

Data Description: Reason for failure

NIDS: OAuth2 client has been registered successfully (002e0036)

This event is generated when you select the **OAuth Client Applications** option under **Novell Audit Logging** on the Logging page of an Identity Server configuration.

Description: NIDS: OAuth2 client has been registered successfully

Originator (B): Schema Title: Originator

Data Description: JCC Device ID (AMDEVICEID#device_id:)

Target (U): Schema Title: User Identifier

Data Description: Authentication User Name

SubTarget (Y): null

Text1 (S): Schema Title: Registered At

Data Description: Client registered time stamp in Milliseconds

Text2 (T): Schema Title: Client Name Data Description: Client Name

Text31 (F): Schema Title: Client ID

Data Description: Client ID

NIDS: OAuth2 client has been modified successfully (002e0037)

This event is generated when you select the **OAuth Client Applications** option under **Novell Audit Logging** on the Logging page of an Identity Server configuration.

Description: NIDS: OAuth2 client has been modified successfully

Originator (B): Schema Title: Originator

Data Description: JCC Device ID (AMDEVICEID#device_id:)

Target (U): Schema Title: User Identifier

Data Description: Authentication User Name

SubTarget (Y): null

Text1 (S): Schema Title: Modified At

Data Description: Client modify time stamp in Milliseconds

Text2 (T): Schema Title: Client Name

Data Description: Client Name

Text31 (F): Schema Title: Client ID Description: Client ID

NIDS: OAuth2 client has been deleted successfully (002e0038)

This event is generated when you select the **OAuth Client Applications** option under **Novell Audit Logging** on the Logging page of an Identity Server configuration.

Description: NIDS: OAuth2 client has been deleted successfully

Originator (B): Schema Title: Originator

Data Description: JCC Device ID (AMDEVICEID#device_id:)

Target (U): Schema Title: User Identifier

Data Description: Authentication User Name

SubTarget (Y): null

Text1 (S): Schema Title: Removed At

Data Description: Client deleted time stamp in Milliseconds

Text2 (T): Schema Title: Client Name

Data Description: Client Name

Text31 (F): Schema Title: Client ID Description: Client ID

NIDS: OAuth2 user has provided consent (002e0039)

This event is generated when you select the **OAuth Consent Provided** option under **Novell Audit Logging** on the Logging page of an Identity Server configuration.

Description: NIDS: OAuth2 user has provided consent

Originator (B): Schema Title: Originator

Data Description: JCC Device ID (AMDEVICEID#device_id:)

Target (U): Schema Title: User Identifier

Data Description: Authentication User Name

SubTarget (Y): null

Text1 (S): Schema Title: Provided At

Data Description: Consent provided time stamp in Milliseconds

Text2 (T): null

Text31 (F): null

NIDS: OAuth2 user has revoked consent (002e0040)

This event is generated when you select the **OAuth Consent Revoked** option under **Novell Audit Logging** on the Logging page of an Identity Server configuration.

Description: NIDS: OAuth2 user has revoked consent

Originator (B): Schema Title: Originator

Data Description: JCC Device ID (AMDEVICEID#device_id:)

Target (U): Schema Title: User Identifier

Data Description: Authentication User Name

SubTarget (Y): null

Text1 (S): Schema Title: Revoked At

Data Description: Consent revoked time stamp in Milliseconds

Text2 (T): null

Text31 (F): null

NIDS: OAuth2 token validation success (002e0041)

This event is generated when you select the **OAuth & OpenID Token Validation Success** option under **Novell Audit Logging** on the Logging page of an Identity Server configuration.

Description: NIDS: OAuth2 token validation success

Originator (B): Schema Title: Originator

Data Description: JCC Device ID (AMDEVICEID#device_id:)

Target (U): Schema Title: User Identifier

Data Description: Authentication User Name

SubTarget (Y): null

Text1 (S): Schema Title: Validated At

Data Description: Validated time stamp in Milliseconds

Text2 (T): null

Text31 (F): Schema Title: Expires

Data Description: Expires in seconds

NIDS: OAuth2 token validation failed (002e0042)

This event is generated when you select the **OAuth & OpenID Token Validation Failed** option under **Novell Audit Logging** on the Logging page of an Identity Server configuration.

Description: NIDS: OAuth2 token validation failed

Originator (B): Schema Title: Originator

Data Description: JCC Device ID (AMDEVICEID#device_id:)

Target (U): null

SubTarget (Y): null

Text1 (S): Schema Title: Validated At

Data Description: Validated time stamp in Milliseconds

Text2 (T): null

Text31 (F): Schema Title: Reason

Data Description: Validation failure reason

Data (D): Schema Title: Client IP Address

Description: IP Address of the host from which the token received

NIDS: OAuth2 client registration failed (002e0043)

This event is generated when you select the **OAuth Client Applications** option under **Novell Audit Logging** on the Logging page of an Identity Server configuration.

Description: NIDS: OAuth2 client registration failed

Originator (B): Schema Title: Originator

Data Description: JCC Device ID (AMDEVICEID#device_id:)

Target (U): Schema Title: User Identifier

Data Description: Authentication User Name

SubTarget (Y): null

Text1 (S): Schema Title: Failed At

Data Description: Client registration failed time stamp in Milliseconds

Text2 (T): Schema Title: Client Name

Data Description: Client Name

Text31 (F): Schema Title: Reason

Data Description: Reason for failure

NIDS: Roles PEP Configured (002e0300)

This event is generated for Identity Server roles.

Description: NIDS: Roles PEP Configured

Originator (B): Schema Title: Originator Data Description: JCC Device ID (AMDEVICEID#device_id:)

Target (U): null

SubTarget (Y): null

Text1 (S): null

Text2 (T): null

Text3 (F): null

Value1 (1): 0

Group (G): 0

Data Length (X): Schema Title: Policy Enforcement List Length Data Description: Byte length of PEL

Data (D): Schema Title: Policy Enforcement List Data Description: Policy Enforcement List (PEL) data

Access Gateway: PEP Configured (002e0301)

This event is generated when you enable auditing.

Description: Access Gateway: policy enforcement point (PEP) configured

Originator (B): Schema Title: Originator Data Description: JCC Device ID (AMDEVICEID#device_id:)

Target (U): null

SubTarget (Y): null

Text1 (S): Schema Title: Event Identifier Data Description: Event Tracking Identifier

Text2 (T): null

Text3 (F): null

Value1 (1): Schema Title: Audit Enabled Data Description: 0 = No; 1 = Yes

Group (G): 0

Data Length (X): Schema Title: Policy Enforcement List Length Data Description: Byte length of PEL

Data (D): Schema Title: Policy Enforcement List Data Description: Policy Enforcement List (PEL) data

Roles Assignment Policy Evaluation (002e0320)

This event is generated when you enable auditing.

Description: Roles assignment policy evaluation

Originator (B): Schema Title: Originator Data Description: JCC Device ID (AMDEVICEID#device_id:)

Target (U): null

SubTarget (Y): null

Text1 (S): Schema Title: Authentication Identifier Data Description: IDP Session ID (AMAUTHID#auth_id:)

Text2 (T): Schema Title: Assigned Roles Data Description: Assigned Role or error message

Text3 (F): Schema Title: Policy Action Data Description: Policy Action FDN

Value1 (1): 0

Group (G): 0

Data Length (X): 0

Data (D): null

Access Gateway: Authorization Policy Evaluation (002e0321)

This event is generated when you enable auditing.

Description: Access Gateway: Authorization policy evaluation

Originator (B): Schema Title: Originator Data Description: JCC Device ID (AMDEVICEID#device_id:)

Target (U): null

SubTarget (Y): null

Text1 (S): Schema Title: Authentication Identifier Data Description: IDP Session ID (AMAUTHID#auth_id:)

Text2 (T): Schema Title: Event Identifier Data Description: Event Tracking Identifier

Text3 (F): Schema Title: Policy Action Data Description: Policy Action FDN

Value1 (1): 0

Group (G): 0

Data Length (X): 0

Data (D): null

Access Gateway: Form Fill Policy Evaluation (002e0322)

This event is generated when you enable auditing.

Description: Access Gateway: Form Fill policy evaluation

Originator (B): Schema Title: Originator Data Description: JCC Device ID (AMDEVICEID#device_id:)

Target (U): null

SubTarget (Y): null

Text1 (S): Schema Title: Authentication Identifier Data Description: IDP Session ID (AMAUTHID#auth_id:)

Text2 (T): Schema Title: Event Identifier Data Description: Event Tracking Identifier

Text3 (F): Schema Title: Policy Action Data Description: Policy Action FDN

Value1 (1): 0

Group (G): 0

Data Length (X): 0

Data (D): null

Access Gateway: Identity Injection Policy Evaluation (002e0323)

This event is generated when you enable auditing.

Description: Access Gateway: Identity Injection policy evaluation

Originator (B): Schema Title: Originator Data Description: JCC Device ID (AMDEVICEID#device_id:)

Target (U): null

SubTarget (Y): null

Text1 (S): Schema Title: Authentication Identifier Data Description: IDP Session ID (AMAUTHID#auth_id:)

Text2 (T): Schema Title: Event Identifier Data Description: Event Tracking Identifier

Text3 (F): Schema Title: Policy Action Data Description: Policy Action FDN

Value1 (1): 0

Group (G): 0

Data Length (X): 0

Data (D): null

Access Gateway: Access Denied (0x002e0505)

This event is generated when you select the **Access Denied** option on the Novell Audit page of an Access Gateway.

Description: Access Gateway: Access Denied

In the Event list (**Auditing and Logging** > **Logging Server Options** > **[Name of Novell Audit Secure Logging Server]** > **Novell Access Manager** > **Events**), this column is called **Event Name**.

In a query, this column is called **EventID**.

Event ID: 0x002e0505

Originator (B): Schema Title: Originator Data Description: JCC Device ID (AMDEVICEID#device_id:)

Target (U): Schema Title: Protected Resource Name Data Description: Configured Name of Protected Resource

SubTarget (Y): Schema Title: Protected Resource URL Data Description: Protected Resource URL

Text1 (S): Schema Title: User Identifier Data Description: User DN

Text2 (T): Schema Title: Authentication Identifier Data Description: IDP Session ID (AMAUTHID#auth_id:)

Text3 (F): Schema Title: Event Identifier Data Description: Event Tracking Identifier

Value1 (1): Schema Title: Source IP Address Data Description: User IP address (numeric format – host order)

Group (G): 0

Data Length (X): 0

Data (D): null

Access Gateway: URL Not Found (0x002e0508)

This event is generated when you select the **URL Not Found** option on the Novell Audit page of an Access Gateway.

Description: Access Gateway: URL Not Found

In the Event list (**Auditing and Logging** > **Logging Server Options** > **[Name of Novell Audit Secure Logging Server]** > **Novell Access Manager** > **Events**), this column is called **Event Name**.

In a query, this column is called **EventID**.

Event ID: 0x002e0508

Originator (B): Schema Title: Originator Data Description: JCC Device ID (AMDEVICEID#device_id:)

Target (U): null

SubTarget (Y): Schema Title: Protected Resource URL Data Description: Protected Resource URL

Text1 (S): Schema Title: User Identifier Data Description: User DN

Text2 (T): Schema Title: Authentication Identifier Data Description: IDP Session ID (AMAUTHID#auth_id:)

Text3 (F): Schema Title: Event Identifier Data Description: Event Tracking Identifier

Value1 (1): Schema Title: Source IP Address Data Description: User IP address (numeric format – host order)

Group (G): 0

Data Length (X): 0

Data (D): null

Access Gateway: System Started (0x002e0509)

This event is generated when you select the **System Started** option on the Novell Audit page of an Access Gateway.

Description: Access Gateway: System Started

In the Event list (**Auditing and Logging** > **Logging Server Options** > **[Name of Novell Audit Secure Logging Server]** > **Novell Access Manager** > **Events**), this column is called **Event Name**.

In a query, this column is called **EventID**.

Event ID: 0x002e0509

Originator (B): Schema Title: Originator Data Description: JCC Device ID (AMDEVICEID#device_id:)

Target (U): null

SubTarget (Y): null

Text1 (S): null

Text2 (T): null

Text3 (F): null

Value1 (1): 0

Group (G): 0

Data Length (X): 0

Data (D): null

Access Gateway: System Shutdown (0x002e050a)

This event is generated when you select the **System Shutdown** option on the Novell Audit page of an Access Gateway.

Description: Access Gateway: System Shutdown

In the Event list (**Auditing and Logging** > **Logging Server Options** > **[Name of Novell Audit Secure Logging Server]** > **Novell Access Manager** > **Events**), this column is called **Event Name**.

In a query, this column is called **EventID**.

Event ID: 0x002e050a

Originator (B): Schema Title: Originator Data Description: JCC Device ID (AMDEVICEID#device_id:)

Target (U): null

SubTarget (Y): null

Text1 (S): null

Text2 (T): null

Text3 (F): null

Value1 (1): 0

Group (G): 0

Data Length (X): 0

Data (D): null

Access Gateway: Identity Injection Parameters (0x002e050c)

This event is generated when you select the **Identity Injection Parameters** option on the Novell Audit page of an Access Gateway.

Description: Access Gateway: Identity Injection Parameters

In the Event list (**Auditing and Logging** > **Logging Server Options** > **[Name of Novell Audit Secure Logging Server]** > **Novell Access Manager** > **Events**), this column is called **Event Name**.

In a query, this column is called **EventID**.

Event ID: 0x002e050c

Originator (B): Schema Title: Originator Data Description: JCC Device ID (AMDEVICEID#device_id:)

Target (U): null

SubTarget (Y): Schema Title: Protected Resource URL Data Description: Protected Resource URL

Text1 (S): Schema Title: User Identifier Data Description: User DN

Text2 (T): Schema Title: Authentication Identifier Data Description: IDP Session ID (AMAUTHID#auth_id:)

Text3 (F): Schema Title: Event Identifier Data Description: Event Tracking Identifier

Value1 (1): Schema Title: Injection Location Data Description: 2710 – Auth Header 2720 – Custom Header 2730 – Query Parameters

Group (G): 0

Data Length (X): 0

Data (D): null

Access Gateway: Identity Injection Failed (0x002e050d)

This event is generated when you select the **Identity Injection Failed** option on the Novell Audit page of an Access Gateway.

Description: Access Gateway: Identity Injection Failed

In the Event list (**Auditing and Logging** > **Logging Server Options** > **[Name of Novell Audit Secure Logging Server]** > **Novell Access Manager** > **Events**), this column is called **Event Name**.

In a query, this column is called **EventID**.

Event ID: 0x002e050d

Originator (B): Schema Title: Originator Data Description: JCC Device ID (AMDEVICEID#device_id:)

Target (U): null

SubTarget (Y): Schema Title: Protected Resource URL Data Description: Protected Resource URL

Text1 (S): Schema Title: User Identifier Data Description: User DN

Text2 (T): Schema Title: Authentication Identifier Data Description: IDP Session ID (AMAUTHID#auth_id:)

Text3 (F): Schema Title: Event Identifier Data Description: Event Tracking Identifier

Value1 (1): Schema Title: Injection Location Data Description: 2710 – Auth Header 2720 – Custom Header 2730 – Query Parameters

Group (G): 0

Data Length (X): 0

Data (D): null

Access Gateway: Form Fill Authentication (0x002e050e)

This event is generated when you select the **Form Fill Success** option on the Novell Audit page of an Access Gateway.

Description: Access Gateway: Form Fill Authentication

In the Event list (**Auditing and Logging** > **Logging Server Options** > **[Name of Novell Audit Secure Logging Server]** > **Novell Access Manager** > **Events**), this column is called **Event Name**.

In a query, this column is called **EventID**.

Event ID: 0x002e050e

Originator (B): Schema Title: Originator Data Description: JCC Device ID (AMDEVICEID#device_id:)

Target (U): Schema Title: Protected Resource Name Data Description: Configured name of protected resource

SubTarget (Y): Schema Title: Protected Resource URL Data Description: Protected Resource URL

Text1 (S): Schema Title: User Identifier Data Description: User DN

Text2 (T): Schema Title: Authentication Identifier Data Description: IDP Session ID (AMAUTHID#auth_id:)

Text3 (F): Schema Title: Event Identifier Data Description: Event Tracking Identifier

Value1 (1): 0

Group (G): 0

Data Length (X): 0

Data (D): null

Access Gateway: Form Fill Authentication Failed (0x002e050f)

This event is generated when you select the **Form Fill Failed** option on the Novell Audit page of an Access Gateway.

Description: Access Gateway: Form Fill Authentication Failed

In the Event list (**Auditing and Logging** > **Logging Server Options** > **[Name of Novell Audit Secure Logging Server]** > **Novell Access Manager** > **Events**), this column is called **Event Name**.

In a query, this column is called **EventID**.

Event ID: 0x002e050f

Originator (B): Schema Title: Originator Data Description: JCC Device ID (AMDEVICEID#device_id:)

Target (U): Schema Title: Protected Resource Name Data Description: Configured name of protected resource

SubTarget (Y): Schema Title: Protected Resource URL Data Description: Protected Resource URL

Text1 (S): Schema Title: User Identifier Data Description: User DN

Text2 (T): Schema Title: Authentication Identifier Data Description: IDP Session ID (AMAUTHID#auth_id:)

Text3 (F): Schema Title: Event Identifier Data Description: Event Tracking Identifier

Value1 (1): 0

Group (G): 0

Data Length (X): 0

Data (D): null

Access Gateway: URL Accessed (0x002e0512)

This event is generated when you select the **URL Accessed** option on the Novell Audit page of an Access Gateway.

Description: Access Gateway: URL Accessed

In the Event list (**Auditing and Logging** > **Logging Server Options** > **[Name of Novell Audit Secure Logging Server]** > **Novell Access Manager** > **Events**), this column is called **Event Name**.

In a query, this column is called **EventID**.

Event ID: 0x002e0512

Originator (B): Schema Title: Originator Data Description: JCC Device ID (AMDEVICEID#device_id:)

Target (U): null

SubTarget (Y): Schema Title: Protected Resource URL Data Description: Protected Resource URL

Text1 (S): Schema Title: User Identifier Data Description: User DN

Text2 (T): Schema Title: Authentication Identifier Data Description: IDP Session ID (AMAUTHID#auth_id:)

Text3 (F): Schema Title: Event Identifier Data Description: Event Tracking Identifier

Value1 (1): Schema Title: Source IP Address Data Description: User IP address (numeric format – host order)

Group (G): 0

Data Length (X): 0

Data (D): null

Access Gateway: IP Access Attempted (0x002e0513)

This event is generated when you select the **IP Access Attempted** option on the Novell Audit page of an Access Gateway.

Description: Access Gateway: IP Access Attempted

In the Event list (**Auditing and Logging** > **Logging Server Options** > **[Name of Novell Audit Secure Logging Server]** > **Novell Access Manager** > **Events**), this column is called **Event Name**.

In a query, this column is called **EventID**.

Event ID: 0x002e0513

Originator (B): Schema Title: Originator Data Description: JCC Device ID (AMDEVICEID#device_id:)

Target (U): null

SubTarget (Y): Schema Title: Protected Resource URL Data Description: Protected Resource URL

Text1 (S): Schema Title: User Identifier Data Description: User DN

Text2 (T): Schema Title: Authentication Identifier Data Description: IDP Session ID (AMAUTHID#auth_id:)

Text3 (F): Schema Title: Event Identifier Data Description: Event Tracking Identifier

Value1 (1): Schema Title: Source IP Address Data Description: User IP address (numeric format – host order)

Group (G): 0

Data Length (X): 0

Data (D): null

Access Gateway: Webserver Down (0x002e0515)

This event is generated when you select the **IP Access Attempted** option on the Novell Audit page of an Access Gateway.

Description: Access Gateway: One of the Web servers is not reachable

In the Event list (**Auditing and Logging** > **Logging Server Options** > **[Name of Novell Audit Secure Logging Server]** > **Novell Access Manager** > **Events**), this column is called **Event Name**.

In a query, this column is called **EventID**.

Event ID: 0x002e0515

Originator (B): Schema Title: Originator Data Description: JCC Device ID (AMDEVICEID#device_id:)

Target (U): null

SubTarget (Y): null

Text1 (S): WebServer hostname

Text2 (T): null

Text3 (F): null

Value1 (1): WebServer IP Address

Group (G): 0

Data Length (X): 0

Data (D): null

Access Gateway: All WebServers for a Service is Down (0x002e0516)

This event is generated when you select the IP Access Attempted option on the Novell Audit page of an Access Gateway.

Description: Access Gateway: All Web servers for a service are down

In the Event list (**Auditing and Logging** > **Logging Server Options** > **[Name of Novell Audit Secure Logging Server]** > **Novell Access Manager** > **Events**), this column is called **Event Name**.

In a query, this column is called **EventID**.

Event ID: 0x002e0516

Originator (B): Schema Title: Originator Data Description: JCC Device ID (AMDEVICEID#device_id:)

Target (U): null

SubTarget (Y): null

Text1 (S): WebServer Hostname

Text2 (T): null

Text3 (F): null

Value1 (1): WebServer IP address

Group (G): 0

Data Length (X): 0

Data (D): null

Management Communication Channel: Health Change (0x002e0601)

This event is generated when you select the **Health Changes** option on the Access Manager Auditing page.

Description: Management Communication Channel: Health Change

In the Event list (**Auditing and Logging** > **Logging Server Options** > **[Name of Novell Audit Secure Logging Server]** > **Novell Access Manager** > **Events**), this column is called **Event Name**.

In a query, this column is called **EventID**.

Event ID: 0x002e0601

Originator (B): Schema Title: Originator Data Description: “devmanagement” (AMDEVICEID#devmanagement:)

Target (U): null

SubTarget (Y): null

Text1 (S): Schema Title: Changed Device Data Description: IP address and device type of the changed device

Text2 (T): Schema Title: Old State Data Description: Old State

Text3 (F): Schema Title: New State Data Description: New State

Value1 (1): 0

Group (G): 0

Data Length (X): 0

Data (D): null

Management Communication Channel: Device Imported (0x002e0602)

This event is generated when you select the **Server Imports** option on the Access Manager Auditing page.

Description: Management Communication Channel: Device Imported

In the Event list (**Auditing and Logging** > **Logging Server Options** > **[Name of Novell Audit Secure Logging Server]** > **Novell Access Manager** > **Events**), this column is called **Event Name**.

In a query, this column is called **EventID**.

Event ID: 0x002e0602

Originator (B): Schema Title: Originator Data Description: “devmanagement” (AMDEVICEID#devmanagement:)

Target (U): null

SubTarget (Y): null

Text1 (S): Schema Title: Device Data Description: IP address and device type of the changed device

Text2 (T): blank string

Text3 (F): blank string

Value1 (1): 0

Group (G): 0

Data Length (X): 0

Data (D): null

Management Communication Channel: Device Deleted (0x002e0603)

This event is generated when you select the **Server Deletes** option on the Access Manager Auditing page.

Description: Management Communication Channel: Device Deleted

In the Event list (**Auditing and Logging** > **Logging Server Options** > **[Name of Novell Audit Secure Logging Server]** > **Novell Access Manager** > **Events**), this column is called **Event Name**.

In a query, this column is called **EventID**.

Event ID: 0x002e0603

Originator (B): Schema Title: Originator Data Description: “devmanagement” (AMDEVICEID#devmanagement:)

Target (U): null

SubTarget (Y): null

Text1 (S): Schema Title: Device Data Description: IP address and device type of the changed device

Text2 (T): Schema Title: Administrator Data Description: DN of the administrator deleting the device

Text3 (F): blank string

Value1 (1): 0

Group (G): 0

Data Length (X): 0

Data (D): null

Management Communication Channel: Device Configuration Changed (0x002e0604)

This event is generated when you select the **Configuration Changes** option on the Access Manager Auditing page.

Description: Management Communication Channel: Device Configuration Changed

In the Event list (**Auditing and Logging** > **Logging Server Options** > **[Name of Novell Audit Secure Logging Server]** > **Novell Access Manager** > **Events**), this column is called **Event Name**.

In a query, this column is called **EventID**.

Event ID: 0x002e0604

Originator (B): Schema Title: Originator Data Description: “devmanagement” (AMDEVICEID#devmanagement:)

Target (U): null

SubTarget (Y): null

Text1 (S): Schema Title: Device Data Description: IP address and device type of the changed device

Text2 (T): Schema Title: Administrator Data Description: DN of the administrator invoking the configuration change

Text3 (F): blank string

Value1 (1): 0

Group (G): 0

Data Length (X): 0

Data (D): null

Management Communication Channel: Device Alert (0x002e0605)

This event is generated when you enable auditing.

Description: Management Communication Channel: Device Alert

In the Event list (**Auditing and Logging** > **Logging Server Options** > **[Name of Novell Audit Secure Logging Server]** > **Novell Access Manager** > **Events**), this column is called **Event Name**.

In a query, this column is called **EventID**.

Event ID: 0x002e0605

Originator (B): Schema Title: Originator Data Description: “devmanagement” (AMDEVICEID#devmanagement:)

Target (U): null

SubTarget (Y): null

Text1 (S): Schema Title: Device Data Description: IP address of the device generating the alert

Text2 (T): Schema Title: Alert Message Data Description: alert message string

Text3 (F): blank string

Value1 (1): 0

Group (G): 0

Data Length (X): 0

Data (D): null

Risk-Based Authentication: 002e0025

This event is generated when you select the **Risk-Based Authentication Succeeded** option under Novell Audit Logging on the Logging page of an Identity Server configuration.

Description: Risk-Based additional authentication executed successfully for user.

Originator (B): Schema Title: Originator Data Description: JCC Device ID (AMDEVICEID#device_id:)

Target (U): Schema Title: User Identifier Data Description: User DN

SubTarget (Y): Schema Title: Authentication Identifier Description: IDP Session ID (AMAUTHID#auth_id:)

Text1 (S): Schema Title: RiskScore Description: Risk score(number).

Text2 (T): Schema Title: RiskLevel Description: Risk category defined by risk score value.

Text3 (F): Schema Title: Additional authentication class Description: Additional Authentication class name executed as part of risk based authentication.

Value1 (1): 0

Group (G): 0

Data Length (X): 0

Data (D): Schema Title: Client IP Address Description: IP Address of the host from which the authentication succeeded.

Risk-Based Authentication: 002e0026

This event is generated when you select the **Risk-Based Authentication Failed** option under Novell Audit Logging on the Logging page of an Identity Server configuration.

Description: Risk-Based authentication failed for user.

Originator (B): Schema Title: Originator Data Description: JCC Device ID (AMDEVICEID#device_id:)

Target (U): Schema Title: User Identifier Data Description: User DN

SubTarget (Y): Schema Title: Authentication Identifier Description: IDP Session ID (AMAUTHID#auth_id:)

Text1 (S): Schema Title: RiskScore Description: Risk score(number).

Text2 (T): Schema Title: RiskLevel Description: Risk category defined by risk score value.

Text3 (F): Schema Title: Additional authentication class Description: Additional Authentication class name executed as part of risk based authentication.

Value1 (1): 0

Group (G): 0

Data Length (X): 0

Data (D): Schema Title: Client IP Address Description: IP Address of the host from which the authentication succeeded.

Risk-Based Authentication: 002e0027

This event is generated when you select the **Risk-Based Authentication Action Invoked** option under Novell Audit Logging on the Logging page of an Identity Server configuration.

Description: Risk-Based authentication action for user.

Originator (B): Schema Title: Originator Data Description: JCC Device ID (AMDEVICEID#device_id:)

Target (U): Schema Title: User Identifier Data Description: User DN

SubTarget (Y): Schema Title: Authentication Identifier Description: IDP Session ID (AMAUTHID#auth_id:)

Text1 (S): Schema Title: RiskScore Description: Risk score(number).

Text2 (T): Schema Title: RiskLevel Description: Risk category defined by risk score value.

Text3 (F): Schema Title: Action taken Description: Risk category defined action taken as part of risk based authentication.

Value1 (1): 0

Group (G): 0

Data Length (X): 0

Data (D): Schema Title: Client IP Address Description: IP Address of the host from which the authentication succeeded.

Event Codes

Event codes for Access Manager Appliance consist of 4 fields that describe the type of code and the module that produced it:

- Severity (1 digit)
  - 1 = severe - Describes problems that needs to be resolved in order for the system to run correctly.
  - 2 = error - Describes that a failure occurred, but the system is operational.
  - 3 = warn - Describes a situation that may exist that the administrator should be aware of and may need to address. The system is currently running properly
  - 4 = config - Describes configuration related information.
  - 5 = info - Describes events that occur.
  - 6 = debug - Describes execution points within the software.
  - 9 = internal - Describes an error that is for internal use only. This error code will not be documented in any public documentation.
- Component issuing the error code (3 digits)
- Sub-grouping for further classification within a component (2 digits)
- Event code (three digits)

0	000	00	000
Severity	**Component field**	**Sub-grouping**	**Event Code**

The following sections divide the event codes by component, then describe them:

- “Administration Console (009)” on page 1085
- “Identity Server (001)” on page 1122
- “Linux Access Gateway Appliance(045)” on page 1163
- “Access Gateway Service (046)” on page 1164
- “Policy Engine (008)” on page 1168
- “SOAP Policy Enforcement Point (011)” on page 1173
- “Backup and Restore (010)” on page 1178
- “NetIQ Modular Authentication Class (012)” on page 1183

Administration Console (009)

Component 009

- Subgroup 01: Certificate Manager
- Subgroup 02: Application
- Subgroup 03: Platform
- Subgroup 04: Web UI
- Subgroup 05: Roma Application
- Subgroup 06: Policy

Event Code	*Description*	**Remedy**
	Application	
100901001	Error getting web manager.	**Cause:** The Administration Console was not installed correctly or has become corrupt. **Action:** Verify installation.
100901002	Error in initializing the dirCerts APIs.	**Cause:** The Administration Console was not installed correctly or has become corrupt. Specifically, the PKI and/or certificate management jars may be missing or have mismatched versions. **Action:** Verify that the `certmgr.jar` file is contained in the `/var/opt/novell/tomcat4/webapps/roma/WEB-INF/lib` directory and that PKI has been installed. Verify that the Java command line contains the following: `-Djava.library.path=/opt/novell/lib` Verify that `npki.jar` is in the classpath.
100901003	Error in init.	**Cause:** The Administration Console was not installed correctly or has become corrupt. **Action:** Verify installation.
100901004	Error in CertHandler.getMultipartParamValue.	**Cause:** Servlet error when retrieving data from a multipart form. **Action:** Submit log to Novell Support for analysis and resolution.
100901008	Could not remove certificate with the given alias from the keystore.	**Cause:** The keystore that contains the certificate might not exist or might have become corrupt. **Action:** View the configuration store and find the keystore object and check that the certificate is no longer in the key list. If it is there, manually remove it. Also, find the keystore on the file system of the device and remove the key manually, using the Java keytool program for JKS keystores.

Event Code	*Description*	Remedy
100901010	Error In CertHandler.doGetSigningCertDN.	**Cause:** Unable to retrieve the DN of the signing cert. **Cause:** The signing cert does not exist. **Cause:** The signing keystore does not exist. **Action:** View the Identity Server Configuration's Signing keystore to verify that it exists and contains a certificate. If the signing keystore does not exist, there has been an error during the import of an Identity Server or during the creation of an Identity Server Configuration. Check to make sure that there are no corrupt Identity Server configurations. If the signing keystore does exist, add or replace a certificate.
100901011	Error in creating or configuring one or more of the Identity Server Configuration cluster keystores.	**Cause:** Test certificates might have been accidentally deleted from the file system. **Cause:** Error communicating with the Identity Server(s) while pushing down the test certificates. **Action:** Use the exception stack trace to discover a more detailed description of the error. Go to the Certificates tab and verify that the test-connector, test-signing, test-encryption, test-provider, test-consumer certificates have not been deleted. Also verify they still exist on the file system. Go to the Trusted Roots tab and verify that the configCA trusted root has not been deleted and that it exists in the configuration store. These test certificates are pushed down to each Identity Server during the creation of an Identity Server configuration. You can delete the Identity Server configuration and create a new one and add the Identity Servers back into the new configuration.
100901012	keystore already exists.	**Cause:** You are trying to create a keystore that already exists on the device. **Action:** Use the existing keystore.
100901013	Error in init (using reflection to call a method has failed in init).	**Cause:** The java class is unable to locate another java class through reflection. **Action:** Submit log to Novell Support for analysis and resolution.

Event Code	*Description*	Remedy
700901014	Cannot add non-existent key to keystore.	**Cause:** The certificate you are trying to add to a keystore does not exist. **Action:** Specify a valid key to be added to the keystore.
700901015	Cannot add key to non-existent keystore.	**Cause:** The keystore does not exist. **Action:** Specify a valid keystore or create the keystore.
700901016	Could not add key to keystore because the alias was too long.	**Cause:** Some platforms and keystore formats only support a limited number of characters in the alias name. **Action:** Use a shorter alias.
700901017	Could not add key to keystore because the maximum number of keys has been reached.	**Cause:** Many keystores allow only one key to be contained in it because the keystore has a specific purpose in Access Manager Appliance. **Action:** Remove unused keys from the keystore and try again.
700901020	Cannot remove non-existent key from keystore.	**Cause:** The key no longer exists in Access Manager Appliance. **Action:** View the configuration store and find the keystore object and manually remove the key from the key list.
700901021	Cannot remove key from non-existent keystore.	**Cause:** The keystore does not exist. **Action:** Specify a valid keystore.
100901023	CertHandler.doGetCertFromServer: Could not connect to server IP and port.	**Cause:** The server IP or DNS name and port combination is not reachable. **Action:** Verify that the IP address or DNS name exists and that the port is correct. You can try connecting to it with a web browser or other utility.
100901024	CertHandler.doGetCertFromServer: certificate was not obtained from server IP and port.	**Cause:** The server IP or DNS name and port combination had no certificate to be presented. **Action:** Verify that the IP address or DNS name exists and that the port is correct. Verify that the server you are attempting to import the certificate from has a certificate. You can try connecting to it with a web browser or other utility.

Event Code	*Description*	Remedy
100901025	Error in handleException.	**Cause:** The exception reported has no details associated with it. **Action:** Scroll up in the log to see if there is a stack trace immediately above this error, determine what steps you had taken to create this error condition, and submit the log and steps to Novell Support.
100901026	The node keystore does not exist. Cannot add cluster keys to a non-existent keystore.	**Cause:** The grouping of Identity Servers (Identity Server Configuration) or Access Gateways is trying to locate a keystore on one of the Identity Server or Access Gateway devices but the keystore cannot be found. **Action:** Verify that the Identity Servers and Access Gateway devices had no errors during import to the Administration Console. Try to re-import the devices.
100901027	Error in CertHandler.getNIDPDeviceKeystoreName (The name of the device's keystore was not found).	**Cause:** The cluster keystore representation object was not found. **Cause:** The cluster keystore representation did not have a device type specified. **Action:** Delete and recreate the Identity Server Configuration or Access Gateway Group that is causing the problem and then re-add the members.
100901028	Error in CertHandler.isTomcatCert (Unable to determine if the specified certificate is the one being used by Tomcat).	**Cause:** The certificate representation has missing or invalid attributes. **Action:** Delete this certificate and re-import it.
100901030	Error in CertHandler.getNodeKeystoreNames (The cluster object was not found in the configuration store, or the cluster server list was empty).	**Cause:** The cluster object was not found in the configuration store, the type of the cluster could not be determined, or the cluster server list was empty. **Action:** No action needed unless your devices are unable to communicate. If you are having problems with communication, delete and recreate the Identity Server configuration or Access Gateway cluster that is causing the problem.
100901031	Error in CertHandler.getClusterDisplayName (The cluster object was not found in the configuration store).	**Action:** Delete and recreate the Identity Server configuration or Access Gateway cluster that is causing the problem and then re-add the members.
100901032	The device does not exist but the certificate is in a keystore assigned to that device.	**Cause:** It's possible the device is in a partially-imported state. **Action:** Delete the keystore, if possible, and re-import the device.

Event Code	*Description*	**Remedy**
100901033	The device does not exist but the keystore is assigned to that device.	**Cause:** It's possible the device is in a partially-imported state. **Action:** Delete the keystore, if possible, and re-import the device.
100901034	Unable to retrieve the primary member of the group.	**Cause:** The group is corrupt. **Action:** Delete the group, re-create it, and re-add the members.
100901035	Unable to remove the node keystore setting off the Access Gateway group device.	**Cause:** Could not locate the keystore object in the configuration store. **Action:** No action required.
700901036	Unable to set the Update Servers status.	**Cause:** Communication error. **Action:** Manually restart or update the device.
700901037	Unable to remove all keys from keystore.	**Cause:** The keystore does not exist. **Cause:** There is a corrupt key in the keystore. **Action:** Manually remove each certificate from the keystore.
700901038	Unable to reinitialize keystore contents for a particular device in a group or configuration.	**Cause:** One of the device keystores does not exist. **Action:** Re-create the keystore or delete and recreate the group or configuration and then re-add the devices to it. **Cause:** There was an error either removing all certificates from a keystore. **Action:** Manually remove all certificates from the keystore and then remove and re-add that device to the group/configuration. **Cause:** There was an error adding the test certificates to a keystore. **Action:** Verify that the test certificates exist (see error 1.009.01.011 for more detail). Manually add the test certificates to the keystore. Or remove the device from the group/configuration and re-add it.
700901039	Unable to assess whether the keystore contains a tomcat connector certificate.	**Cause:** The cluster keystore representation does not exist or is corrupt. **Cause:** Unable to locate the devices in the group/configuration. **Action:** Delete and recreate the group/ configuration and re-add the devices to it.

Event Code	*Description*	Remedy
700901040	Error adding a key to keystore during the renew certificate process.	**Cause:** The original certIficate information could not be located. **Action:** Manually create a new certificate and place it into all the keystores which previously held the certificate being renewed.
100901041	Unable to extract the public key from a key during the auto-import public certificate process.	**Cause:** The source keystore does not exist. **Action:** Select a valid keystore. **Cause:** The specified source key does not exist. **Action:** Verify that the key you have specified to export the public certificate from exists.
100901042	Unable to set up the initial keys for the cluster.	**Cause:** When trying to locate the cluster keystores so that their contents can be initialized, one or more of those keystore representations could not be found. **Action:** Delete and recreate the Identity Server configuration or Access Gateway cluster.
100901043	The source keystore does not exist. Cannot push keys from a non-existent keystore.	**Cause:** The source keystore does not exist. **Action:** Usually the source keystore is a cluster keystore representation. Try deleting and recreating the Identity Server configuration or Access Gateway cluster to ensure those cluster keystore representations get created.
	Application	
100902001	Error - Exception thrown in eventOccurred of vcdn.application.sc.alert.AlertEventListener	**Cause:** Cannot post alert to internal subsystem. **Action:** Non-fatal error. No action required.
100902002	Error - Exception thrown in eventOccurred of vcdn.application.sc.alert.AlertEventListener.	**Cause:** Cannot post alert to internal subsystem. **Action:** Submit the `app_sc.0.log` file for resolution.
100902003	Error - Exception thrown in logAlert of vcdn.application.sc.alert.AlertLogger.	**Cause:** Problem occurred update the Identity Server Alert count. **Action:** Non-fatal error. May be a symptom of a more serious condition. Submit the `app_sc.0.log` file for resolution.

Event Code	*Description*	Remedy
100902004	Error - Exception thrown in the execute method of vcdn.application.sc.alert.CertUpdateWork.	**Cause:** Could not update or read the list of trusted server certificates. **Action:** Be sure the `/var/opt/novell/novlwww/devman.cacerts` file exists, is a valid Java keystore, and is not corrupted. To check its status, enter the following command: `/opt/novell/java/bin/keytool -v -list -keystore devman.cacerts` Otherwise, be sure the config store is running and functioning properly.
100902005	Error - (The specified device) has not been imported. Failed to start device.	**Cause:** The Identity Server was not properly imported. **Action:** Go to Access Gateway Server List and click **Repair Import.** (The repair import. functionality works for any server type.) Otherwise, submit the `app_sc.0.log` file for resolution.
100902006	Error importing device (with the specified ID).	**Cause:** The Server was not properly imported. **Action:** Go to Access Gateway Server List and click **Repair Import.** (The repair import. functionality works for any server type.) If this fails, reinstall the server component.
100902007	Error - Import failed. Retrying.	**Cause:** Unable to communicate with the Server being imported. **Action:** Be sure the firewall is allowing port 1443 traffic. Otherwise allow the system to retry for several minutes. If the server does not appear in the Server List, click **Repair Import** to resolve the issue. Otherwise, submit the `app_sc.0.log` file for resolution.
100902008	Error auto importing. Retry.	**Cause:** Unable to communicate with the Server being imported. **Action:** Be sure the firewall is allowing port 1443 traffic. Otherwise allow the system to retry for several minutes. If the server does not appear in the Server List, click **Repair Import.** to resolve the issue. Otherwise, submit the `app_sc.0.log` file for resolution.
100902009	Error - Could not create subcontext: cn=(The specified Context)	**Cause:** Error creating Server object in config store during import. **Action:** Go to Access Gateway Server List and click **Repair Import**. (The repair import functionality works for any server type.) Otherwise, submit the `app_sc.0.log` file for resolution.

Event Code	*Description*	Remedy
100902010	Error - (The given ESP) does not exist!	**Cause:** There was a error during the Administration Console installation. **Action:** Reinstall the Administration Console.
100902011	Error - Exception reading (the given ESP)	**Cause:** The file required during the import process could not be read. **Action:** Be sure the indicated file can be read by the novlwww user.
100902012	Error - Could not import LDIF.	**Cause:** The error occurred while creating the configuration for the Embedded Service Provider. **Action:** Go to Access Gateway Server List and click **Repair Import.** (The repair import functionality works for any server type.) Otherwise, submit the `app_sc.0.log` file for resolution.
100902013	Error - Could not find (the specified DN)	**Cause:** Error connecting to the config store while importing the Embedded Service Provider. **Action:** Go to Access Gateway Server List and click **Repair Import**. (The repair import functionality works for any server type.) Otherwise, submit the `app_sc.0.log` file for resolution. You might need to restart the Administration Console.
100902014	Error - ESP Configuration was not found, so auto-import failed.	**Cause:** Could not find the configuration for the imported Embedded Service Provider. **Action:** Go to Access Gateway Server List and click **Repair Import**. (The repair import functionality works for any server type.) Otherwise, submit the `app_sc.0.log` file for resolution.
100902015	Error - Exception thrown in importDevice of vcdn.application.sc.alert.RegisterCommand.	**Cause:** Error during import of server component. **Action:** Go to Access Gateway Server List and click **Repair Import**. (The repair import functionality works for any server type.) Otherwise, submit the `app_sc.0.log` file for resolution.
100902016	Error - ImportThread null member vars.	**Cause:** Internal error occurred during import. **Action:** Go to Access Gateway Server List and click **Repair Import**. (The repair import functionality works for any server type.) Otherwise, submit the `app_sc.0.log` file for resolution.

Event Code	*Description*	Remedy
100902017	Error - Could not connect to eDir for certs.	**Cause:** Either the primary Administration Console is down (if this is a secondary console), or the config store is down. **Action:** Be sure the config store is operating properly and that port 554 is not blocked by a firewall.
100902018	Error during execution.	**Cause:** Error executing an external program during import process. **Action:** Go to Access Gateway Server List and click **Repair Import**. (The repair import functionality works for any server type.) Otherwise, submit the `app_sc.0.log` file for resolution.
100902019	Error - Could not get (the given number of) bytes of payload data.	**Cause:** An error occurred while trying to read data for a command. **Action:** Ensure the server component is operating properly. Otherwise, submit the `app_sc.0.log` file for resolution.
100902020	Error - VException thrown while executing command in vcdn.application.sc.alert.AlertCommandHandler.	**Cause:** Problem executing a command from a server component. **Action:** Ensure the server component is operating properly. Otherwise, submit the `app_sc.0.log` file for resolution.
100902021	Error - VCDNException thrown in performConfiguration of vcdn.application.sc.config.AGApplyWork	**Cause:** Problem occurred while sending configuration to Access Gateway server. **Action:** Ensure the server component is operating properly. Otherwise, submit the `app_sc.0.log` file for resolution.
100902022	Error - VCDNException thrown in responseReceived method of vcdn.application.sc.config.AGApplyWork	**Cause:** Error occurred in processing the response from an Access Gateway server. **Action:** Ensure the server component is operating properly. Otherwise, submit the `app_sc.0.log` file for resolution.
100902023	Error - VCDNException thrown in performConfiguration method of vcdn.application.sc.config.AGConfigWork	**Cause:** Error occurred while sending configuration to Access Gateway server. **Action:** Ensure the server component is operating properly. Otherwise, submit the `app_sc.0.log` file for resolution.
100902024	Error - VCDNException thrown in responseReceived method of vcdn.application.sc.config.AGConfigWork	**Cause:** Error occurred in processing the response from an Access Gateway server. **Action:** Ensure the server component is operating properly. Otherwise, submit the `app_sc.0.log` file for resolution.

Event Code	*Description*	Remedy
100902025	Error - Exception thrown in processAGResponse method of vcdn.application.sc.config.AGConfigWork	**Cause:** Error occurred in processing the response from an Access Gateway server. **Action:** Ensure the server component is operating properly. Otherwise, submit the `app_sc.0.log` file for resolution.
100902031	Error - SchedulerException thrown in configureDeviceNow method of vcdn.application.sc.config.ConfigManager	**Cause:** Error occurred while scheduling an immediate apply of the current configuration. **Action:** Submit the `app_sc.0.log` file for resolution.
100902032	Error - Exception thrown in the execute method of vcdn.application.sc.config.ConfigWork	**Cause:** Error occurred while performing pending actions. **Action:** Submit the `app_sc.0.log` file for resolution.
100902033	Error setting LDAP attribute in performPendingActions of vcdn.application.sc.config.ConfigWork	**Cause:** Pending actions could not be completed because of a problem communicating with the config store. **Action:** Ensure the config store is functioning correctly. Otherwise, submit the `app_sc.0.log` file for resolution.
100902034	Error invoking method in performPendingActions of vcdn.application.sc.config.ConfigWork	**Cause:** Problem occurred while invoking a method during a pending action. **Action:** Submit the `app_sc.0.log` file for resolution.
100902035	Error executing pending action (name) in performPendingActions of vcdn.application.sc.config.ConfigWork	**Cause:** Problem occurred while displaying a pending dialog message. **Action:** This is a non-fatal error. If the problem persists, submit the `app_sc.0.log` file for resolution.
100902036	Error - Exception thrown in getConfigXML of vcdn.application.sc.config.ConfigWork	**Cause:** Error occurred while retrieving XML data from the config store. **Action:** Ensure the config store is functioning correctly. Otherwise, submit the `app_sc.0.log` file for resolution.
100902037	Error - VException thrown in saveInDB method of vcdn.application.sc.config.ConfigWork	**Cause:** Error occurred while saving the applied configuration in the config store. **Action:** Ensure the config store is functioning correctly. Otherwise, submit the `app_sc.0.log` file for resolution.
100902038	Error - VException thrown in configFinished method of vcdn.application.sc.config.DeviceConfigApplyWork	**Cause:** Error occurred while sending the Audit event for a changed configuration. **Action:** Ensure the Audit server and the config store are functioning properly. Otherwise, submit the `app_sc.0.log` file for resolution.

Event Code	*Description*	Remedy
100902039	Error - VException thrown in configFinished method of vcdn.application.sc.config.DeviceConfigWork	**Cause:** Error occurred while sending the Audit event for a changed configuration. **Action:** Ensure the Audit server and the config store are functioning properly. Otherwise, submit the `app_sc.0.log` file for resolution.
100902040	Error - Exception thrown in processConfigDiff method of vcdn.application.sc.config.DeviceGroupConfigWork	**Cause:** Error occurred while parsing the XML for a group configuration. **Action:** Error occurred while sending the Audit event for a changed configuration. **Action:** Submit the `app_sc.0.log` file for resolution.
100902041	Error - Exception thrown in memberConfigFinished method of vcdn.application.sc.config.DeviceGroupConfigWork	**Cause:** Error occurred while processing a group member configuration apply response. **Action:** Ensure the server component is functioning properly. Otherwise, submit the `app_sc.0.log` file for resolution.
100902042	Error - Exception thrown in removePendingFromFailedList method of vcdn.application.sc.config.DeviceGroupConfigWork	**Cause:** Error occurred while re-applying a server configuration. **Action:** Submit the `app_sc.0.log` file for resolution.
100902043	Error - SchedulerException thrown in scheduleMultiDeviceWorks method of vcdn.application.sc.config.DeviceGroupConfigWork	**Cause:** Error occurred while scheduling a group configuration. **Action:** Submit the `app_sc.0.log` file for resolution.
100902044	Error - Exception thrown in the execute method of vcdn.application.sc.config.DeviceGroupConfigWork	**Cause:** Error occurred while scheduling a group configuration. **Action:** Submit the `app_sc.0.log` file for resolution.
100902045	Error - VException thrown in performWork method of vcdn.application.sc.config.MultiDeviceConfigWork	**Cause:** Error occurred while applying configuration to a group member. **Action:** Ensure the server component is functioning properly. Otherwise, submit the `app_sc.0.log` file for resolution.
100902046	Error - Exception thrown in performWork method of vcdn.application.sc.config.MultiDeviceConfigWork	**Cause:** Error occurred while applying configuration to a group member. **Action:** Ensure the server component is functioning properly. Otherwise, submit the `app_sc.0.log` file for resolution.
100902047	Error - SchedulerException thrown in getDeviceGroupConfigWork method of vcdn.application.sc.config.MultiDeviceConfigWork	**Cause:** Error occurred while trying to get the scheduled configuration. **Action:** Ensure the config store is functioning correctly. Otherwise, submit the `app_sc.0.log` file for resolution.

Event Code	*Description*	Remedy
100902048	Error - VException thrown in configFinished method of vcdn.application.sc.config.MultiDeviceConfigWork	**Cause:** Error occurred while importing status from a group member. **Action:** Ensure the config store is functioning correctly. Otherwise, submit the `app_sc.0.log` file for resolution.
100902049	Error - VCDNException thrown in the execute method of vcdn.application.sc.command.AGCommandWork	**Cause:** Error occurred while sending a command to an Access Gateway server. **Action:** Ensure the server component is functioning correctly. Otherwise, submit the `app_sc.0.log` file for resolution.
100902050	Error - Exception thrown in the sendCommand method of vcdn.application.sc.command.AGCommandWork	**Cause:** Error occurred while sending a command to an Access Gateway server. **Action:** Ensure the server component is functioning correctly. Otherwise, submit the `app_sc.0.log` file for resolution.
100902051	Error - Exception thrown in the processAGResponse method of vcdn.application.sc.command.AGCommandWork	**Cause:** Error occurred while processing a command response from an Access Gateway server. **Action:** Ensure the server component is functioning correctly. Otherwise, submit the `app_sc.0.log` file for resolution.
100902055	Error - IOException thrown in the addCommand method of vcdn.application.sc.command.CertCommand	**Cause:** Error generating certificate command. **Action:** Submit the `app_sc.0.log` file for resolution.
100902056	Error - IOException thrown in the generateCmd method of vcdn.application.sc.command.CertCommand	**Cause:** Error generating certificate command. **Action:** Submit the `app_sc.0.log` file for resolution.
100902057	Error - IOException thrown in the setCertChainData method of vcdn.application.sc.command.CertCommand	**Cause:** Error generating chained certificate command. **Action:** Submit the `app_sc.0.log` file for resolution.
100902058	Error - VCDNException thrown in the execute method of vcdn.application.sc.command.IDPCommandWork	**Cause:** Error occurred while sending a command to an Identity Server ESP server. **Action:** Ensure the server component is functioning correctly. Otherwise, submit the `app_sc.0.log` file for resolution.
100902059	Error - VCDNException thrown in the sendCommand method of vcdn.application.sc.command.IDPCommandWork	**Cause:** Error occurred while sending a command to an Identity Server or ESP server. **Action:** Ensure the server component is functioning correctly. Otherwise, submit the `app_sc.0.log` file for resolution.

Event Code	*Description*	Remedy
100902060	Error - NamingException thrown in the updateNIDPCommandStatus method of vcdn.application.sc.command.IDPCommandWork	**Cause:** Error occurred while processing a command response from an Identity Server or ESP. **Action:** Ensure the server component is functioning correctly. Otherwise, submit the `app_sc.0.log` file for resolution.
010090261	Error - VException thrown in the updateNIDPCommandStatus method of vcdn.application.sc.command.IDPCommandWork	**Cause:** Error occurred while processing a command response from an Identity Server or ESP. **Action:** Ensure the server component is functioning correctly. Otherwise, submit the `app_sc.0.log` file for resolution.
100902062	Error - Exception thrown in the processIDPResponse method of vcdn.application.sc.command.IDPCommandWork	**Cause:** Error occurred while processing a command response from an Identity Server or ESP. **Action:** Ensure the server component is functioning correctly. Otherwise, submit the `app_sc.0.log` file for resolution.
100902063	Error - VCDNException thrown in the execute method of vcdn.application.sc.command.JCCCommandWork	**Cause:** Error occurred while executing a server command. **Action:** Ensure the server component is functioning correctly. Otherwise, submit the `app_sc.0.log` file for resolution.
100902064	Error - Exception thrown in the sendCommand method of vcdn.application.sc.command.JCCCommandWork	**Cause:** Error occurred while sending a server command. **Action:** Ensure the server component is functioning correctly. Otherwise, submit the `app_sc.0.log` file for resolution.
100902065	Error - Exception thrown in the processResponse method of vcdn.application.sc.command.JCCCommandWork	**Cause:** Error occurred while processing a response from a server command. **Action:** Ensure the server component is functioning correctly. Otherwise, submit the `app_sc.0.log` file for resolution.
300902069	Exception changing factory LocalAddress.	**Cause:** Error occurred while changing factory XML during configuration import. **Action:** Submit the `app_sc.0.log` file for resolution.
100902070	Error - ConverterException thrown in the getCurrentDeviceXML method of vcdn.application.sc.core.AGDevice	**Cause:** Error occurred during translation of NetWare Access Gateway configuration. **Action:** Submit the `app_sc.0.log` file for resolution.

Event Code	*Description*	Remedy
100902071	Error - NamingException thrown in the importDevice method of vcdn.application.sc.core.AGDevice	**Cause:** Config store could not be accessed or an internal error occurred. **Action:** Ensure the config store is functioning properly. Otherwise, submit the `app_sc.0.log` file for resolution.
100902072	Error - VException thrown in the importDevice method of vcdn.application.sc.core.AGDevice	**Cause:** Config store could not be accessed or an internal error occurred. **Action:** Ensure the config store is functioning properly. Otherwise, submit the `app_sc.0.log` file for resolution.
100902073	Error - Exception thrown in the importDevice method of vcdn.application.sc.core.AGDevice	**Cause:** Config store could not be accessed or an internal error occurred. **Action:** Ensure the config store is functioning properly. Otherwise, submit the `app_sc.0.log` file for resolution.
100902074	Error - NamingException thrown in the vcdn.application.sc.core.AuditManager constructor.	**Cause:** Config store could not be accessed or an internal error occurred. **Action:** Ensure the config store is functioning properly. Otherwise, submit the `app_sc.0.log` file for resolution.
100902075	Error - JDOMException thrown in the processDocument method of vcdn.application.sc.core.AuditManager	**Cause:** Audit XML data could not be parsed. **Action:** Submit the `app_sc.0.log` file for resolution.
100902076	Error - Exception thrown in the processDocument method of vcdn.application.sc.core.AuditManager	**Cause:** Invalid data format. **Action:** Attempt the operation again. Otherwise, submit the `app_sc.0.log` file for resolution.
100902077	Error - Exception thrown in the setDefaultServer method of vcdn.application.sc.core.AuditManager	**Cause:** Config store could not be accessed or an internal error occurred. **Action:** Ensure the config store is functioning properly. Otherwise, submit the `app_sc.0.log` file for resolution.
100902078	Error - VException thrown in the writeConfig method of vcdn.application.sc.core.AuditManager	**Cause:** Config store could not be accessed or an internal error occurred. **Action:** Ensure the config store is functioning properly. Otherwise, submit the `app_sc.0.log` file for resolution.
100902079	Error - NamingException thrown in the writeConfig method of vcdn.application.sc.core.AuditManager	**Cause:** Config store could not be accessed or an internal error occurred. **Action:** Ensure the config store is functioning properly. Otherwise, submit the `app_sc.0.log` file for resolution.

Event Code	*Description*	Remedy
100902080	Error - Exception thrown in the writeConfig method of vcdn.application.sc.core.AuditManager	**Cause:** Config store could not be accessed or an internal error occurred. **Action:** Ensure the config store is functioning properly. Otherwise, submit the `app_sc.0.log` file for resolution.
100902081	Error - SException thrown in the getIDPConfigObject method of vcdn.application.sc.core.AuditManager	**Cause:** Config store could not be accessed or an internal error occurred. **Action:** Ensure the config store is functioning properly. Otherwise, submit the `app_sc.0.log` file for resolution.
100902082	Error - NamingException thrown in the getIDPConfigObject method of vcdn.application.sc.core.AuditManager	**Cause:** Config store could not be accessed or an internal error occurred. **Action:** Ensure the config store is functioning properly. Otherwise, submit the `app_sc.0.log` file for resolution.
100902083	Error - Exception thrown in the getIDPConfigObject method of vcdn.application.sc.core.AuditManager	**Cause:** Config store could not be accessed or an internal error occurred. **Action:** Ensure the config store is functioning properly. Otherwise, submit the `app_sc.0.log` file for resolution.
100902084	Error - NullPointerException thrown in the logEvent method of vcdn.application.sc.core.AuditManager	**Cause:** Error logging Novell Audit event. **Action:** Ensure the Novell Audit server is functioning properly. Otherwise, submit the `app_sc.0.log` file for resolution.
100902085	Error - Exception thrown in the creatElement method of vcdn.application.sc.core.DeviceConfig	**Cause:** Internal XML error. **Action:** Submit the `app_sc.0.log` file for resolution.
100902086	Error - Exception thrown in the setLastModified method of vcdn.application.sc.core.DeviceConfig	**Cause:** Internal XML error. **Action:** Submit the `app_sc.0.log` file for resolution.
300902087	Warning - Exception thrown in the getLastScheduledWorkID method of vcdn.application.sc.core.DeviceGroupManager	**Cause:** The last executed command status ID could not be read. **Action:** Non-fatal error.
100902088	Error - Could not get version from device. Make sure it is running properly.	**Cause:** Could not get version from device. **Action:** Make sure the server component is running properly, then click **Repair Import** to resolve the issue. Otherwise, submit the `app_sc.0.log` file for resolution.
100902089	Error - NamingException thrown in the importDevice method of vcdn.application.sc.core.DeviceManager	**Cause:** Error importing device. **Action:** Make sure the server component is running properly, then click **Repair Import** to resolve the issue. Otherwise, submit the `app_sc.0.log` file for resolution.

Event Code	*Description*	Remedy
100902090	Error - VException thrown in the importDevice method of vcdn.application.sc.core.DeviceManager	**Cause:** Error importing device. **Action:** Make sure the server component is running properly, then click **Repair Import** to resolve the issue. Otherwise, submit the `app_sc.0.log` file for resolution.
100902091	Error - InvocationTargetException thrown in the importDevice method of vcdn.application.sc.core.DeviceManager	**Cause:** Error importing device. **Action:** Make sure the server component is running properly, then click **Repair Import** to resolve the issue. Otherwise, submit the `app_sc.0.log` file for resolution.
100902092	Error - Exception thrown in the importDevice method of vcdn.application.sc.core.DeviceManager	**Cause:** Error importing device. **Action:** Make sure the server component is running properly, then click **Repair Import** to resolve the issue. Otherwise, submit the `app_sc.0.log` file for resolution.
100902093	Error - Could not find esp cfg SCC to remove in cluster container.	**Cause:** Error deleting improperly imported server. **Action:** Non-fatal error.
100902094	Error deleting the trusted IDP entry for ESP.	**Cause:** Error accessing config store. **Action:** Ensure the config store is functioning properly. Otherwise, submit the `app_sc.0.log` file for resolution.
100902095	Error - NamingException thrown in the setHealthCheck method of vcdn.application.sc.core.DeviceManager	**Cause:** Error saving health status in config store. **Action:** Ensure the config store is functioning properly. Otherwise, submit the `app_sc.0.log` file for resolution.
100902096	Error - Could not find the DN specified.	**Cause:** Error saving health status in config store. **Action:** Ensure the server component imported correctly and the config store is functioning properly. Otherwise, submit the `app_sc.0.log` file for resolution.
100902097	Error - Exception thrown in the deleteDevice method of vcdn.application.sc.core.DeviceManager	**Cause:** Error occurred while deleting the server objects. **Action:** Ensure the config store is functioning properly. Otherwise, submit the `app_sc.0.log` file for resolution.
100902098	Error - Exception thrown in the setHealthCheck method of vcdn.application.sc.core.DeviceManager	**Cause:** Error updating the version following an upgrade of a server component. **Action:** Allow the operation to try again. Otherwise, submit the `app_sc.0.log` file for resolution.

Event Code	*Description*	Remedy
300902099	Warning - Exception thrown in the getLastScheduledWorkID method of vcdn.application.sc.core.DeviceManager	**Cause:** The last executed command status ID could not be read. **Action:** Non-fatal error.
300902100	Device is not imported.	**Cause:** Server component is sending health to Administration console that does not recognize the server. **Action:** Click **Repair Import** to resolve the issue. Otherwise, submit the `app_sc.0.log` file for resolution.
300902101	Identity configuration not found for device.	**Cause:** Identity server configuration not found in config store. **Action:** Non-fatal error.
100902102	Error - Exception thrown in the createCertEntry method of vcdn.application.sc.core.KeyManager	**Cause:** The config store is not reachable or the user does not have rights to modify the config store **Action:** Verify the config store is up and that the user has rights to create objects in the following container: `ou=KeyContainer,ou=Partition,ou=PartitionsContainer,ou=VCDN_root,ou=accessManagerContainer,o=novell`
100902103	Error - Exception thrown in the deleteCertEntry method of vcdn.application.sc.core.KeyManager	**Cause:** The config store is not reachable or the user does not have rights to modify the config store **Action:** Verify the config store is up and that the user has rights to delete objects in the following container: `ou=KeyContainer,ou=Partition,ou=PartitionsContainer,ou=VCDN_root,ou=accessManagerContainer,o=novell`
100902104	Error - Exception thrown in the modifyCertEntryXml method of vcdn.application.sc.core.KeyManager	**Cause:** The config store is not reachable or the user does not have rights to modify the config store **Action:** Verify the config store is up and that the user has rights to modify objects in the following container: `ou=KeyContainer,ou=Partition,ou=PartitionsContainer,ou=VCDN_root,ou=accessManagerContainer,o=novell`

Event Code	*Description*	Remedy
100902105	Error - Exception thrown in the createKeyStoreEntry method of vcdn.application.sc.core.KeyManager	**Cause:** The config store is not reachable or the user does not have rights to modify the config store **Action:** Verify the config store is up and that the user has rights to create objects in the following container: `ou=KeyContainer,ou=Partition,ou=PartitionsContainer,ou=VCDN_root,ou=accessManagerContainer,o=novell`
100902106	Error - Exception thrown in the deleteKeyStoreEntry method of vcdn.application.sc.core.KeyManager	**Cause:** The config store is not reachable or the user does not have rights to modify the config store **Action:** Verify the config store is up and that the user has rights to delete objects in the following container: `ou=KeyContainer,ou=Partition,ou=PartitionsContainer,ou=VCDN_root,ou=accessManagerContainer,o=novell`
100902107	Error - Exception thrown in the modifyKeyStoreEntryXml method of vcdn.application.sc.core.KeyManager	**Cause:** The config store is not reachable or the user does not have rights to modify the config store **Action:** Verify the config store is up and that the user has rights to modify objects in the following container: `ou=KeyContainer,ou=Partition,ou=PartitionsContainer,ou=VCDN_root,ou=accessManagerContainer,o=novell`
100902108	Error - Exception thrown in the createElement method of vcdn.application.sc.core.PolicyConfig	**Cause:** Error creating an element in the specified XML document. **Action:** Submit the `app_sc.0.log` file for resolution.
100902109	Error - Exception thrown in the setLastModified method of vcdn.application.sc.core.PolicyConfig	**Cause:** Error setting an attribute value on modified elements. **Action:** Submit the `app_sc.0.log` file for resolution.
100902113	Error - Exception thrown in the sendData method of vcdn.application.sc.core.work.DeleteDeviceWork	**Cause:** Error communicating with component. **Action:** Ensure the server component is functioning correctly. Otherwise, submit the `app_sc.0.log` file for resolution.
100902114	Error - Exception thrown in the execute method of vcdn.application.sc.core.work.ReimportDeviceWork	**Cause:** Error occurred while executing a server command. **Action:** Ensure the server component is functioning correctly. Otherwise, submit the `app_sc.0.log` file for resolution.

Event Code	*Description*	Remedy
100902115	Error - Exception thrown in the getHealth method of vcdn.application.sc.health.HealthCheck	**Cause:** Error occurred while executing a server command. **Action:** Ensure the server component is functioning correctly. Otherwise, submit the `app_sc.0.log` file for resolution.
100902116	Error - Inner Exception thrown in the execute method of vcdn.application.sc.health.HealthCheck	**Cause:** Error occurred while executing a server command. **Action:** Ensure the server component is functioning correctly. Otherwise, submit the `app_sc.0.log` file for resolution.
100902117	Error - Outer Exception thrown in the execute method of vcdn.application.sc.health.HealthCheck	**Cause:** Error occurred while executing a server command. **Action:** Ensure the server component is functioning correctly. Otherwise, submit the `app_sc.0.log` file for resolution.
100902118	Error - VException thrown in the eventOccurred method of vcdn.application.sc.health.HealthEventListener	**Cause:** Error occurred while receiving/ logging a health event. **Action:** Ensure the server component is functioning correctly. Otherwise, submit the `app_sc.0.log` file for resolution.
100902119	Error getting Health Module or Service	**Cause:** Error occurred while executing a server command. **Action:** Ensure the server component is functioning correctly. Otherwise, submit the `app_sc.0.log` file for resolution.
100902120	Error - Exception thrown in the execute method of vcdn.application.sc.health.HealthUpdateWork	**Cause:** Error occurred while executing a server command. **Action:** Ensure the server component is functioning correctly. Otherwise, submit the `app_sc.0.log` file for resolution.
	Platform	
100903001	Error - Unable to find a trusted client certificate.	**Cause:** There was a problem during the import of the device. **Action:** Consult the documentation to re-import the device into the Administration Console.
100903002	Error building delayed response.	**Cause:** Error occurred while processing a request. **Action:** Ensure the server component is functioning correctly. Otherwise, submit the `app_sc.0.log` file for resolution.

Event Code	*Description*	Remedy
100903003	Error setting return code in HttpServletResponse.	**Cause:** Error occurred while processing a request. **Action:** Ensure the server component is functioning correctly. Otherwise, submit the `app_sc.0.log` file for resolution.
100903004	Error - DelayedResponseListener thread failed to start.	**Cause:** Error occurred while processing a delayed response. **Action:** Ensure the server component is functioning correctly. Otherwise, submit the `app_sc.0.log` file for resolution.
100903005	Error in the ResponseHandler thread of the DelayedResponseListener.	**Cause:** Error occurred while processing a response. **Action:** Ensure the server component is functioning correctly. Otherwise, submit the `app_sc.0.log` file for resolution.
100903006	Error creating XML Element in ResponseBuilder.	**Cause:** Error occurred while editing XML. **Action:** Ensure the server component is functioning correctly. Otherwise, submit the `app_sc.0.log` file for resolution.
100903007	Error waiting on mutex in RequestDispatcher.	**Cause:** Error occurred while getting responses. **Action:** Ensure the server component is functioning correctly. Otherwise, submit the `app_sc.0.log` file for resolution.
100903008	Error notifying mutex in RequestDispatcher.	**Cause:** Error occurred while receiving a response. **Action:** Ensure the server component is functioning correctly. Otherwise, submit the `app_sc.0.log` file for resolution.
100903009	Error receiving in SendInternal of VConnection.	**Cause:** Error occurred while receiving an internal response. **Action:** Ensure the server component is functioning correctly. Otherwise, submit the `app_sc.0.log` file for resolution.
100903010	Error getting response code in VConnection.	**Cause:** Error occurred while getting the code. **Action:** Ensure the server component is functioning correctly. Otherwise, submit the `app_sc.0.log` file for resolution.
100903011	Error in stopScheduledResponses of VConnection.	**Cause:** Error occurred while attempting to stop scheduled responses. **Action:** Ensure the server component is functioning correctly. Otherwise, submit the `app_sc.0.log` file for resolution.

Event Code	*Description*	Remedy
100903012	Error in ConsumeData of VConnection.	**Cause:** Error occurred while reading data. **Action:** Ensure the server component is functioning correctly. Otherwise, submit the `app_sc.0.log` file for resolution.
100903013	Error in sendData of VConnection.	**Cause:** Error occurred while sending data. **Action:** Ensure the server component is functioning correctly. Otherwise, submit the `app_sc.0.log` file for resolution.
100903014	Error in getHeaders of VConnection.	**Cause:** Error occurred while getting headers. **Action:** Ensure the server component is functioning correctly. Otherwise, submit the `app_sc.0.log` file for resolution.
100903015	Error in receive of VConnection.	**Cause:** Error occurred while receiving a response. **Action:** Ensure the server component is functioning correctly. Otherwise, submit the `app_sc.0.log` file for resolution.
	Web UI	
100904001	Error reading manager data in UIManager.	**Cause:** Error occurred while reading data. **Action:** Ensure the server component is functioning correctly. Otherwise, submit the `app_sc.0.log` file for resolution.
100904002	Error during auto authentication in WebApplicaitonFilter.	**Cause:** Error occurred while authenticating. **Action:** Ensure the server component is functioning correctly. Otherwise, submit the `app_sc.0.log` file for resolution.
100904003	Error - Exception thrown in doFilter of WebApplicationFilter.	**Cause:** Error getting panel data. **Action:** Ensure the server component is functioning correctly. Otherwise, submit the `app_sc.0.log` file for resolution.
100904004	Error - Exception thrown in logout of WebApplicationFilter.	**Cause:** Error occurred while logging out. **Action:** Ensure the server component is functioning correctly. Otherwise, submit the `app_sc.0.log` file for resolution.
100904005	Error - VException thrown in getUserInfo of WebManager.	**Cause:** Error occurred while getting user information. **Action:** Ensure the server component is functioning correctly. Otherwise, submit the `app_sc.0.log` file for resolution.

Event Code	*Description*	Remedy
100904006	Error - Exception thrown in getDeviceInfo of WebManager.	**Cause:** Error occurred while getting device information. **Action:** Ensure the server component is functioning correctly. Otherwise, submit the `app_sc.0.log` file for resolution.
100904007	Error - Exception thrown in getPolicyInfo of WebManager.	**Cause:** Error occurred while getting policy information. **Action:** Ensure the server component is functioning correctly. Otherwise, submit the `app_sc.0.log` file for resolution.
100904008	Error - Exception thrown in getTypeSpecificationInfo of WebManager.	**Cause:** Error occurred while getting policy type specification information. **Action:** Ensure the server component is functioning correctly. Otherwise, submit the `app_sc.0.log` file for resolution.
100904009	Error - Exception thrown in getDeviceConfig of WebManager.	**Cause:** Error occurred while getting device configuration. **Action:** Ensure the server component is functioning correctly. Otherwise, submit the `app_sc.0.log` file for resolution.
100904010	Error - Exception thrown in getPolicyConfig of WebManager.	**Cause:** Error occurred while getting device configuration. **Action:** Ensure the server component is functioning correctly. Otherwise, submit the `app_sc.0.log` file for resolution.
100904011	Error - Exception thrown in getTypeSpecificationConfig of WebManager.	**Cause:** Error occurred while getting policy type specification configuration. **Action:** Ensure the server component is functioning correctly. Otherwise, submit the `app_sc.0.log` file for resolution.
100904012	Error - Exception thrown in parameterMapToString of WebManager.	**Cause:** Error occurred while getting parameter information. **Action:** Ensure the server component is functioning correctly. Otherwise, submit the `app_sc.0.log` file for resolution.
100904013	Error while logging out user {0}.	**Cause:** Error occurred while logging out NDS user object. **Action:** Ensure the server component is functioning correctly. Otherwise, submit the `app_sc.0.log` file for resolution.
100904014	Error - Exception thrown in getSelectionCriteria of WebPanel.	**Cause:** Error occurred while getting selection criteria. **Action:** Ensure the server component is functioning correctly. Otherwise, submit the `app_sc.0.log` file for resolution.

Event Code	*Description*	Remedy
100904015	Error - Exception thrown in getPanelVersion of WebPanel.	**Cause:** Error occurred while getting panel version. **Action:** Ensure the server component is functioning correctly. Otherwise, submit the `app_sc.0.log` file for resolution.
100904016	Error - Group Config failed.	**Cause:** Error occurred while applying group configuration. **Action:** Ensure the server component is functioning correctly. Otherwise, submit the `app_sc.0.log` file for resolution.
100904017	Error - Schedule Group Config failed.	**Cause:** Error occurred while scheduling group configuration. **Action:** Ensure the server component is functioning correctly. Otherwise, submit the `app_sc.0.log` file for resolution.
100904018	Error - Update XML and Device Config failed.	**Cause:** Error occurred while updating configuration. **Action:** Ensure the server component is functioning correctly. Otherwise, submit the `app_sc.0.log` file for resolution.
100904019	Error - Unlock Config failed.	**Cause:** Error occurred while unlocking the configuration. **Action:** Ensure the server component is functioning correctly. Otherwise, submit the `app_sc.0.log` file for resolution.
100904020	Error - Exception thrown in do_cancelPendingConfig of ConfigWorkDispatcher.	**Cause:** Error occurred while canceling a pending configuration. **Action:** Ensure the server component is functioning correctly. Otherwise, submit the `app_sc.0.log` file for resolution.
100904021	Error - Exception thrown in do_cancelPendingConfig of ConfigWorkDispatcher.	**Cause:** Error occurred while canceling a pending configuration. **Action:** Ensure the server component is functioning correctly. Otherwise, submit the `app_sc.0.log` file for resolution.
100904022	Error - Exception thrown in do_reapplyPendingConfig of ConfigWorkDispatcher.	**Cause:** Error occurred while reapplying a pending configuration. **Action:** Ensure the server component is functioning correctly. Otherwise, submit the `app_sc.0.log` file for resolution.
100904023	Error - Exception thrown in do_deviceConfig of ConfigWorkDispatcher.	**Cause:** Error occurred while applying configuration. **Action:** Ensure the server component is functioning correctly. Otherwise, submit the `app_sc.0.log` file for resolution.

Event Code	*Description*	Remedy
100904024	Error - Exception thrown in do_scheduleDeviceConfig of ConfigWorkDispatcher.	**Cause:** Error occurred while scheduling configuration. **Action:** Ensure the server component is functioning correctly. Otherwise, submit the `app_sc.0.log` file for resolution.
200904025	Error - XML VALIDATION FAILED. PLEASE CHECK APP_SC LOG.	**Cause:** XML created by GUI does not match the XML schema and fails validation. **Action:** Cancel the changes that were made and try again. In any case, submit the `app_sc.0.log` file for resolution.
100904026	Error applying settings in ConfigXmlUpdateDispatcher.	**Cause:** Error occurred while applying configuration. **Action:** Ensure the server component is functioning correctly. Otherwise, submit the `app_sc.0.log` file for resolution.
100904027	Error - Exception thrown in do_save of ConfigXmlUpdateDispatcher.	**Cause:** Error occurred while saving configuration. **Action:** Ensure the server component is functioning correctly. Otherwise, submit the `app_sc.0.log` file for resolution.
100904028	Error - Exception thrown in do_cancel of ConfigXmlUpdateDispatcher.	**Cause:** Error occurred while canceling configuration changes. **Action:** Ensure the server component is functioning correctly. Otherwise, submit the `app_sc.0.log` file for resolution.
100904029	Error - Exception thrown in do_refreshConfig of ConfigXmlUpdateDispatcher.	**Cause:** Error occurred while refreshing configuration manager panel. **Action:** Ensure the server component is functioning correctly. Otherwise, submit the `app_sc.0.log` file for resolution.
100904030	Error - Exception thrown in setLastModParams of ConfigXmlUpdateDispatcher.	**Cause:** Error occurred while setting an XML attribute. **Action:** Ensure the server component is functioning correctly. Otherwise, submit the `app_sc.0.log` file for resolution.
100904031	Error - IOException thrown in getXPathMap of ConfigXmlUpdateDispatcher.	**Cause:** Error occurred while xpath mapping on the current panel. **Action:** Ensure the server component is functioning correctly. Cancel changes on the current panel, return, and try again. Otherwise, submit the `app_sc.0.log` file for resolution.

Event Code	*Description*	Remedy
100904032	Error decoding: {0}.	**Cause:** Error occurred while xpath mapping on the current panel. **Action:** Ensure the server component is functioning correctly. Cancel changes on the current panel, return, and try again. Otherwise, submit the `app_sc.0.log` file for resolution.
100904033	Error - Exception thrown in processRequest of ExceptionDispatcher.	**Cause:** Error occurred while processing request. **Action:** Ensure the server component is functioning correctly. Otherwise, submit the `app_sc.0.log` file for resolution.
100904034	Error - Exception thrown in the service method of ServletDispatcher.	**Cause:** Error occurred while processing request. **Action:** Ensure the server component is functioning correctly. Otherwise, submit the `app_sc.0.log` file for resolution.
100904035	Error - Exception thrown in ServletDispatcher.	**Cause:** Error occurred while inserting dispatchers. **Action:** Ensure the server component is functioning correctly. Otherwise, submit the `app_sc.0.log` file for resolution.
100904036	Error - Exception thrown in processRequest of DeviceCommandHandler.	**Cause:** Error occurred while processing request. **Action:** Ensure the server component is functioning correctly. Otherwise, submit the `app_sc.0.log` file for resolution.
100904037	Error - VException thrown in setNIDPCommandState of DeviceCommandHandler.	**Cause:** Error occurred while accessing data store. **Action:** Ensure the data store is functioning correctly. Otherwise, submit the `app_sc.0.log` file for resolution.
100904038	Error - NamingException thrown in setNIDPCommandState of DeviceCommandHandler.	**Cause:** Error occurred while accessing data store. **Action:** Ensure the data store is functioning correctly. Otherwise, submit the `app_sc.0.log` file for resolution.
100904039	Error - Could not find signing keystore for {0}.	**Cause:** An error occurred during the import of the device. **Action:** Consult the documentation and re-import the device into the Administration Console.

Event Code	*Description*	Remedy
100904040	Error - Could not find encryption keystore for {0}.	**Cause:** An error occurred during the import of the device. **Action:** Consult the documentation and re-import the device into the Administration Console.
100904041	Error - Could not find connector keystore for {0}.	**Cause:** An error occurred during the import of the device. **Action:** Consult the documentation and re-import the device into the Administration Console.
100904042	Error - Could not find trust keystore for {0}.	**Cause:** An error occurred during the import of the device. **Action:** Consult the documentation and re-import the device into the Administration Console.
100904043	Error - Could not find OCSP trust keystore for {0}.	**Cause:** An error occurred during the import of the device. **Action:** Consult the documentation and re-import the device into the Administration Console.
100904044	Error - No keys were assigned to keystore: {0}.	**Cause:** The keystore does not have any certificates in it. This may or may not be a bad condition. For instance, the OCSP trust store can be empty and that should not cause a problem. The signing, encryption, connector, provider, and consumer keystores should have one certificate in them. If it is empty, either the device import failed or the user manually removed the certificate from the keystore. **Action:** Check the keystore using the UI. If the keystore shows that it has a certificate, then the device import probably failed. Consult the documentation and re-import the device and also try deleting and re-creating the NIDP configuration. Also, try replacing the certificate in the keystore through the UI.
100904045	Error - Exception thrown in processRequest of UpgradeDeviceGroupHandler.	**Cause:** Error occurred while processing request. **Action:** Ensure the server component is functioning correctly. Otherwise, submit the `app_sc.0.log` file for resolution.
100904046	Error - Exception thrown in processRequest of UpgradeDeviceHandler.	**Cause:** Error occurred while processing request. **Action:** Ensure the server component is functioning correctly. Otherwise, submit the `app_sc.0.log` file for resolution.

Event Code	*Description*	**Remedy**
100904047	Error - Exception thrown in getUpgradeInfo of UpgradeDeviceHandler.	**Cause:** Error occurred while getting update information. **Action:** Ensure the server component is functioning correctly. Otherwise, submit the `app_sc.0.log` file for resolution.
	Application Handlers	
100905001	Error during repair import.	**Cause:** Error occurred while attempting to repair import. **Action:** Delete the server from the list and reinstall. Otherwise, submit the `app_sc.0.log` file for resolution.
100905002	Error - Failed to remove server.	**Cause:** Error occurred while attempting to remove server. **Action:** Submit the `app_sc.0.log` file for resolution.
100905003	Error setting device groups.	**Cause:** Error occurred while attempting to mark a server as a member of a group. **Action:** Delete the server from the group and retry or delete the group and recreate. Otherwise, submit the `app_sc.0.log` file for resolution.
100905004	Error setting device admin.	**Cause:** Error occurred while attempting to give an Administrator access to a server. **Action:** Submit the `app_sc.0.log` file for resolution.
100905005	Error - Exception thrown while importing appliance.	**Cause:** Error occurred while importing a server. **Action:** Delete the server from the list and reinstall. Otherwise, submit the `app_sc.0.log` file for resolution.
100905006	Error getting health info.	**Cause:** Error occurred while getting health information for a server. **Action:** Ensure the server component and the config store are functioning correctly. Otherwise, submit the `app_sc.0.log` file for resolution.
100905007	Error canceling appliance creation.	**Cause:** Internal error. **Action:** Submit the `app_sc.0.log` file for resolution.
100905008	Error creating new CDN.	**Cause:** Internal error. **Action:** Submit the `app_sc.0.log` file for resolution.

Event Code	*Description*	Remedy
100905009	Error removing CDN.	**Cause:** Internal error. **Action:** Submit the `app_sc.0.log` file for resolution.
100905010	Error creating new Admin.	**Cause:** Internal error. **Action:** Submit the `app_sc.0.log` file for resolution.
100905011	Error while changing the cached device port.	**Cause:** Internal error while processing request. **Action:** Ensure the Management IP Address is correct or edit as needed. Otherwise, submit the `app_sc.0.log` file for resolution.
100905012	Error while changing the cached device password.	**Cause:** Internal error while processing request. **Action:** Ensure the Management Password is correct or edit as needed. Otherwise, submit the `app_sc.0.log` file for resolution.
100905013	Error - Exception thrown while processing request in EditApplianceHandler	**Cause:** Internal error while processing request. **Action:** Ensure all values on the Server Details Edit page are correct and edit as needed. Otherwise, submit the `app_sc.0.log` file for resolution.
100905014	Error - Exception thrown while modifying device handler in EditDeviceHandler.	**Cause:** Error occurred while processing a request. **Action:** Ensure the server component is functioning correctly. Otherwise, submit the `app_sc.0.log` file for resolution.
100905015	Error - Exception thrown while changing password in EditDeviceHandler.	**Cause:** Error occurred while processing a request. **Action:** Ensure the server component is functioning correctly. Otherwise, submit the `app_sc.0.log` file for resolution.
200905016	Error - Exception thrown while editing CDN in EditPublisherHandler.	**Cause:** Internal error. **Action:** Ensure the config store is functioning correctly. Otherwise, submit the `app_sc.0.log` file for resolution.
200905017	Error - Exception thrown while updating CDN in EditPublisherHandler.	**Cause:** Internal error. **Action:** Ensure the config store is functioning correctly. Otherwise, submit the `app_sc.0.log` file for resolution.

Event Code	*Description*	Remedy
200905018	Error - Failed to update the device groups for this user.	**Cause:** Internal error. **Action:** Ensure the config store is functioning correctly. Otherwise, submit the `app_sc.0.log` file for resolution.
200905019	Error - Failed to update the devices for this user.	**Cause:** Internal error. **Action:** Ensure the config store is functioning correctly. Otherwise, submit the `app_sc.0.log` file for resolution.
200905020	Error - Failed to update the cdns for this user.	**Cause:** Internal error. **Action:** Ensure the config store is functioning correctly. Otherwise, submit the `app_sc.0.log` file for resolution.
200905021	Error - Failed to update user data.	**Cause:** Internal error. **Action:** Ensure the config store is functioning correctly. Otherwise, submit the `app_sc.0.log` file for resolution.
100905022	Error processing client certs in GenericPipeHandler.	**Cause:** Internal error while processing request. **Action:** Ensure the server component is functioning correctly. Otherwise, submit the `app_sc.0.log` file for resolution.
200905023	Error accessing XML data item in generic pipe: {0}	**Cause:** Internal error. **Action:** Ensure the config store is functioning correctly. Otherwise, submit the `app_sc.0.log` file for resolution.
200905024	Error parsing XML data item in generic pipe: {0}	**Cause:** Internal error. **Action:** Ensure the config store is functioning correctly. Otherwise, submit the `app_sc.0.log` file for resolution.
200905025	Error processing XML data item in generic pipe: {0}	**Cause:** Internal error. **Action:** Ensure the config store is functioning correctly. Otherwise, submit the `app_sc.0.log` file for resolution.
100905026	Error - Exception thrown in processRequest of GenericPipeHandler: {0}	**Cause:** Internal error. **Action:** Ensure the config store is functioning correctly. Otherwise, submit the `app_sc.0.log` file for resolution.
100905027	Error occurred while creating group {0} : {1}.	**Cause:** Internal error. **Action:** Ensure the config store is functioning correctly or delete the group and recreate it. Otherwise, submit the `app_sc.0.log` file for resolution.

Event Code	*Description*	Remedy
100905028	Error getting device manager in doGroupRemove of GroupCreateHandler.	**Cause:** Internal error. **Action:** Ensure the config store is functioning correctly or delete the group again. Otherwise, submit the `app_sc.0.log` file for resolution.
100905029	Error occurred while removing group {0} : {1}.	**Cause:** Internal error. **Action:** Ensure the config store is functioning correctly or delete the group again. Otherwise, submit the `app_sc.0.log` file for resolution.
100905030	Error occurred while getting device manager in doGroupAlertStatus of GroupCreateHandler.	**Cause:** Unable to get alert status for the group. **Action:** Ensure the config store is functioning correctly. Otherwise, submit the `app_sc.0.log` file for resolution.
100905031	Error occurred while setting alert status for group {0} : {1}.	**Cause:** Unable to set alert status for the group. **Action:** Ensure the config store is functioning correctly. Otherwise, submit the `app_sc.0.log` file for resolution.
100905032	Error occurred while updating group {0} : {1}.	**Cause:** Unable to make updates to the group. **Action:** Ensure the config store is functioning correctly. Otherwise, submit the `app_sc.0.log` file for resolution.
100905033	Error occurred while removing devices from group {0} : {1}.	**Cause:** Unable to remove servers from the group. **Action:** Ensure the config store is functioning correctly. Otherwise, submit the `app_sc.0.log` file for resolution.
100905034	Error - Naming Exception thrown in removeDeviceFromCluster of GroupCreateHandler.	**Cause:** Unable to remove servers from the cluster. **Action:** Ensure the config store is functioning correctly. Otherwise, submit the `app_sc.0.log` file for resolution.
100905035	Error - Exception thrown in removeDeviceFromCluster of GroupCreateHandler.	**Cause:** Error occurred while removing servers from the cluster. **Action:** Ensure the config store is functioning correctly. Otherwise, submit the `app_sc.0.log` file for resolution.
100905036	Error - Exception thrown in removeDeviceFromCluster of GroupCreateHandler.	**Cause:** Error occurred while removing servers from the cluster. **Action:** Ensure the config store is functioning correctly. Otherwise, submit the `app_sc.0.log` file for resolution.

Event Code	*Description*	Remedy
100905037	Error occurred while adding devices to group {0} : {1}.	**Cause:** Error occurred while adding servers to the group. **Action:** Ensure the config store is functioning correctly. Otherwise, submit the `app_sc.0.log` file for resolution.
100905038	Error - Naming Exception thrown in addDeviceToCluster of GroupCreateHandler.	**Cause:** Error occurred while adding servers to the cluster. **Action:** Ensure the config store is functioning correctly. Otherwise, submit the `app_sc.0.log` file for resolution.
100905039	Error - Exception thrown in addDeviceToCluster of GroupCreateHandler.	**Cause:** Error occurred while adding servers to the cluster. **Action:** Ensure the config store is functioning correctly. Otherwise, submit the `app_sc.0.log` file for resolution.
100905040	Error - Exception thrown in addDeviceToCluster of GroupCreateHandler.	**Cause:** Error occurred while adding servers to the cluster. **Action:** Ensure the config store is functioning correctly. Otherwise, submit the `app_sc.0.log` file for resolution.
100905041	Error occurred while adding devices to group {0} : {1}.	**Cause:** Error occurred while adding servers to the cluster. **Action:** Ensure the config store is functioning correctly. Otherwise, submit the `app_sc.0.log` file for resolution.
100905042	Error - VCDNException thrown in processRequest of SyncHandler.	**Cause:** Internal error. **Action:** Ensure the config store is functioning correctly. Otherwise, submit the `app_sc.0.log` file for resolution.
100905043	Error - Exception thrown in processRequest of SyncHandler.	**Cause:** Internal error. **Action:** Ensure the config store is functioning correctly. Otherwise, submit the `app_sc.0.log` file for resolution.
100905044	Error - Exception thrown in modifySystemSync of SyncHandler.	**Cause:** Internal error. **Action:** Ensure the config store is functioning correctly. Otherwise, submit the `app_sc.0.log` file for resolution.
100905045	Error - WSException thrown in isAssignedUser of GroupCreateBean.	**Cause:** Internal error. **Action:** Ensure the config store is functioning correctly. Otherwise, submit the `app_sc.0.log` file for resolution.

Event Code	*Description*	Remedy
100905046	Error - WSException thrown in isAssignedDevice of GroupCreateBean.	**Cause:** Internal error. **Action:** Ensure the config store is functioning correctly. Otherwise, submit the `app_sc.0.log` file for resolution.
100905047	Error - WSException thrown in getApplianceByUrl of GroupCreateBean.	**Cause:** Internal error. **Action:** Ensure the config store is functioning correctly. Otherwise, submit the `app_sc.0.log` file for resolution.
100905048	Error - WSException thrown in generateMembershipList of GroupCreateBean.	**Cause:** Internal error. **Action:** Ensure the config store is functioning correctly. Otherwise, submit the `app_sc.0.log` file for resolution.
100905049	Error - WSException thrown in getAppGroupByName of GroupCreateBean.	**Cause:** Internal error. **Action:** Ensure the config store is functioning correctly. Otherwise, submit the `app_sc.0.log` file for resolution.
100905050	Error - WSException thrown in getDescForThisGroup of GroupCreateBean.	**Cause:** Internal error. **Action:** Ensure the config store is functioning correctly. Otherwise, submit the `app_sc.0.log` file for resolution.
100905051	Error - Exception thrown in getDescForThisGroup of GroupCreateBean.	**Cause:** Internal error. **Action:** Ensure the config store is functioning correctly. Otherwise, submit the `app_sc.0.log` file for resolution.
100905052	Error - WSException thrown in getLastModifiedDate of GroupCreateBean.	**Cause:** Internal error. **Action:** Ensure the config store is functioning correctly. Otherwise, submit the `app_sc.0.log` file for resolution.
100905053	Error - Get appliance groups failed in GroupCreateBean.	**Cause:** Internal error. **Action:** Ensure the config store is functioning correctly or delete group and recreate it. Otherwise, submit the `app_sc.0.log` file for resolution.
100905054	Error - WSException thrown in hasAMembershipIn of GroupCreateBean.	**Cause:** Internal error. **Action:** Ensure the config store is functioning correctly or delete group and recreate it. Otherwise, submit the `app_sc.0.log` file for resolution.
100905055	Error - Get appliances failed in GroupCreateBean.	**Cause:** Internal error. **Action:** Ensure the config store is functioning correctly or delete group and recreate it. Otherwise, submit the `app_sc.0.log` file for resolution.

Event Code	*Description*	Remedy
100905056	Error - Get admins failed in GroupCreateBean.	**Cause:** Internal error. **Action:** Ensure the config store is functioning correctly. Otherwise, submit the `app_sc.0.log` file for resolution.
100905057	Error - WSException thrown in getPerDeviceProperties of GroupCreateBean.	**Cause:** Internal error. **Action:** Ensure the config store is functioning correctly. Otherwise, submit the `app_sc.0.log` file for resolution.
100905058	Error - WSException thrown in getPerUserProperties of GroupCreateBean.	**Cause:** Internal error. **Action:** Ensure the config store is functioning correctly. Otherwise, submit the `app_sc.0.log` file for resolution.
100905059	Error - WSException thrown in getDeviceGroupProperties of GroupCreateBean.	**Cause:** Internal error. **Action:** Ensure the config store is functioning correctly. Otherwise, submit the `app_sc.0.log` file for resolution.
100905060	Error - NamingException thrown in setDeviceClusterConfig of GroupCreateBean.	**Cause:** Internal error. **Action:** Ensure the config store is functioning correctly. Otherwise, submit the `app_sc.0.log` file for resolution.
100905061	Error - Exception thrown in setDeviceClusterConfig of GroupCreateBean.	**Cause:** Internal error. **Action:** Ensure the config store is functioning correctly. Otherwise, submit the `app_sc.0.log` file for resolution.
100905062	Error - VException thrown in clusterServers of GroupCreateBean.	**Cause:** Internal error. **Action:** Ensure the config store is functioning correctly. Otherwise, submit the `app_sc.0.log` file for resolution.
100905063	Error - Exception thrown in clusterServers of GroupCreateBean.	**Cause:** Internal error. **Action:** Ensure the config store is functioning correctly. Otherwise, submit the `app_sc.0.log` file for resolution.
100905064	Error - VException thrown in getAdminList of GroupCreateBean.	**Cause:** Internal error. **Action:** Ensure the config store is functioning correctly. Otherwise, submit the `app_sc.0.log` file for resolution.
100905065	Error - Exception thrown in callRestartESP of SPConfigHandler.	**Cause:** Error occurred while restarting Embedded Service Provider. **Action:** Ensure the server component and ESP are functioning correctly or restart ESP again. Otherwise, submit the `app_sc.0.log` file for resolution.

Event Code	*Description*	Remedy
100905066	Error restarting {0}.	**Cause:** Error occurred while restarting Embedded Service Provider. **Action:** Ensure the server component and ESP are functioning correctly or restart ESP again. Otherwise, submit the `app_sc.0.log` file for resolution.
100905067	Error - Could not lookup {0}.	**Cause:** Error occurred while looking up DN in config store. **Action:** Ensure the config store is functioning correctly. Otherwise, submit the `app_sc.0.log` file for resolution.
100905068	{0}.	**Cause:** Error occurred while accessing config store. **Action:** Ensure the config store is functioning correctly. Otherwise, submit the `app_sc.0.log` file for resolution.
100905069	Error - Exception thrown in createTrustedIDP of SPConfigHandler.	**Cause:** Error occurred while accessing config store. **Action:** Ensure the config store is functioning correctly. Otherwise, submit the `app_sc.0.log` file for resolution.
100905070	Error getting the esp trusted IDP.	**Cause:** Error occurred while accessing config store. **Action:** Ensure the config store is functioning correctly. Otherwise, submit the `app_sc.0.log` file for resolution.
100905071	espTrustAccessDN not set.	**Cause:** Error occurred while accessing config store. **Action:** Ensure the config store is functioning correctly. Otherwise, submit the `app_sc.0.log` file for resolution.
100905072	Error deleting trusted IDP config.	**Cause:** Error occurred while accessing config store. **Action:** Ensure the config store is functioning correctly. Otherwise, submit the `app_sc.0.log` file for resolution.
100905073	Error - VCDNException thrown in processRequest of ScheduleHandler.	**Cause:** Error occurred while processing request. **Action:** Ensure the config store is functioning correctly. Otherwise, submit the `app_sc.0.log` file for resolution.

Event Code	*Description*	Remedy
100905074	Error - Exception thrown in processRequest of ScheduleHandler.	**Cause:** Error occurred while processing request. **Action:** Ensure the config store is functioning correctly. Otherwise, submit the `app_sc.0.log` file for resolution.
100905075	Error - Exception thrown in setEnable of ScheduleHandler.	**Cause:** Error occurred while processing request. **Action:** Ensure the config store is functioning correctly. Otherwise, submit the `app_sc.0.log` file for resolution.
100905076	Error - Exception thrown while removing scheduled work in ScheduleHandler.	**Cause:** Error occurred while processing request. **Action:** Ensure the config store is functioning correctly. Otherwise, submit the `app_sc.0.log` file for resolution.
100905077	Error - Exception thrown while releasing config lock in ScheduleHandler.	**Cause:** Error occurred while unlocking configuration. **Action:** Ensure the config store is functioning correctly. Otherwise, submit the `app_sc.0.log` file for resolution.
100905078	Error - Exception thrown in modify method of ScheduleHandler.	**Cause:** Error occurred while modifying scheduled work. **Action:** Ensure the config store is functioning correctly. Otherwise, submit the `app_sc.0.log` file for resolution.
100905079	Error - Exception thrown in executeNow method of ScheduleHandler.	**Cause:** Error occurred while scheduling work. **Action:** Ensure the config store and server component are functioning correctly. Otherwise, submit the `app_sc.0.log` file for resolution.
100905080	Error - ParamNotFoundException thrown in createSchedule method of ScheduleHandler.	**Cause:** Error occurred while scheduling work. **Action:** Ensure the config store and server component are functioning correctly. Otherwise, submit the `app_sc.0.log` file for resolution.
100905081	Error - Can not forward the request to return page. Nothing can be done.	**Cause:** Internal error. **Action:** Ensure server component is functioning correctly and attempt to navigate to desired panels. Otherwise, submit the `app_sc.0.log` file for resolution.

Event Code	*Description*	**Remedy**
100905082	Error - Exception thrown in create method of ScheduleHandler.	**Cause:** Error occurred while scheduling work. **Action:** Ensure the config store and server component are functioning correctly. Otherwise, submit the `app_sc.0.log` file for resolution.
100905083	Config store Error	**Cause:** The connection to the config store is experiencing problems. **Action:** To diagnose time synchronization issues with multiple Administration Consoles, run the following command on the primary server command-line: `/opt/novell/eDirectory/bin/ndsrepair -T` This will check the overall time synchronization status. If the time is not in sync, then you might want to consider configuring NTP on each server.
	Policy	
1009 06000	Cannot set update status for NULL policy extension.	**Cause:** The composite ID of the extension specified cannot be resolved to an extension ID. **Action:** On the device that is not receiving an Update status, make a configuration change to force the Update link to become active.
1009 06001	Cannot retrieve policy collection info object for the extension.	**Cause:** The extension ID specified cannot be found in the configuration store. **Action:** If you see a problem with your extensions, note this error in the log and call support.
1009 06002	Cannot retrieve device info object for a device	**Cause:** When trying to set the Update status on devices which use an extension, the device info was unable to be located in the configuration store. **Action:** On the device that is not receiving an Update status, make a configuration change to force the Update link to become active.
5009 06000	Attempting to update policy status on devices because the policy extension changed.	**Cause:** Informational message. **Action:** No action necessary.
5009 06001	Setting update policy status for device.	**Cause:** Informational message. **Action:** No action necessary.

Identity Server (001)

Component 001

- Subgroup 01: End user events
- Subgroup 02: Web Service Framework (WSF)
- Subgroup 03: Web Service Consumer (WSC)
- Subgroup 04: User Authentication

Event Code	Message	Remedy
100100001		Type: SEVERE:NIDP:INITIALIZE:001
100100002		Type: SEVERE:NIDP:INITIALIZE:002
100101001	No binding available or set for profile.	Type: SEVERE:NIDP:USERMSG:001 **Cause:** An action using Liberty or SAML protocols could not be completed because the server and trusted provider are not compatibly configured to interact to complete the action. **Action:** Set the desired protocol profiles in the administration tool to match those supported at the trusted provider.
100101043	IDP is unable to load ESP metadata.	Type: SEVERE:NIDP:USERMSG:043 **Cause:** The IDP cannot connect to the metadata URL for the ESP. The IDP may not be able to resolve the domain name for the ESP or if HTTPS is being used, the IDP may not trust the SSL certificate for the ESP. The ESP might also not be running. **Action:** Make sure that certificates for ESP are imported and trusted into IDP configuration. Check the metadata URL for the ESP and make sure the metadata can be retrieved from a browser: http://<DNS_name>/nesp/idff/metadata If you are seeing this error after changing the IP address of the Access Gateway, restart Tomcat on the Identity Server. **Cause:** The IDP needs to have access to the internet to resolve and reach the CRL and OCSP URLs for ESP certificate validation. **Action:** Make sure the internet access is enabled, else the IDP will not trust the ESP certificate even if it has the signing and intermediary certificates.

Event Code	Message	Remedy
100101044	ESP is unable to load IDP metadata	Type: SEVERE:NIDP:USERMSG:044 **Cause:** The ESP cannot connect to the metadata URL for the IDP. The ESP may not be able to resolve the domain name for the IDP or if HTTPS is being used, the ESP may not trust the SSL certificate for the IDP. The IDP may also not be running **Action:** Make sure the IDP is running and that all certificates are imported and trusted. Check the metadata URL for the IDP and make sure the metadata can be retrieved from a browser: http://<DNS_name>/nidp/idff/metadata A common cause is the base URL on the IDP is set incorrectly. For additional help, see "Troubleshooting 100101043 and 100101044 Liberty Metadata Load Errors" on page 982.
100101045	An error happened while the request was being sent to the correct cluster member for processing.	Type: SEVERE:NIDP:USERMSG:045 **Cause:** The target cluster member may be unavailable. **Action:** Ensure that all cluster devices are operating correctly.
100102001	Incomplete web service configuration.	Type: SEVERE:NIDP:WSF:001 **Cause:** The web service instance type (attribute nidsWsfServiceInstanceType on the nidsWsfService object) is not available in the service definition. **Action:** Delete the associated web service definition and recreate it.
100102002	Invalid web service configuration.	Type: SEVERE:NIDP:WSF:002 **Cause:** The web service configuration XML (attribute nidsConfigXML on the nidsWsfService object) has invalid XML. **Action:** Delete the associated web service definition and recreate it.
100102003	Unable to instantiate the web service provider authority class. This class will be com.novell.nidp.liberty.wsf.config.authority.ldap.WSFConfigAuthorityLdap.	Type: SEVERE:NIDP:WSF:003 **Cause:** Some Java error (probably a classpath issue) is causing the main authority class to not instantiate. **Action:** Review how the Access Manager product was installed and attempt to determine if Java class files are being accessed from an unexpected source.
100102004	Unable to load web services.	Type: SEVERE:NIDP:WSF:004 **Cause:** This error catches all failures encountered while trying to load all web services. The reason will be different depending on where the error happened. **Action:** Try to delete and recreate the web services.

Event Code	Message	Remedy
100102005	Unable to access Novell Secret Store.	Type: SEVERE:NIDP:WSF:005 **Cause:** The LDAP connection between the IDP and the User Store must be secure LDAP if Novell Secret Store is to be used as the back end storage for Credential Profile. **Action:** Go to the associated user store and change the connection type to secure LDAP.
100102006	Unable to create user profile object.	Type: SEVERE:NIDP:WSF:006 **Cause:** A Liberty User Profile Object did not exist for the current user, so an attempt was made to create one. That attempt failed! **Action:** Determine if the named container exists and that the administrator user has rights to create objects there.
100102007	Unable to instantiate password callback class.	Type: SEVERE:NIDP:WSF:007 **Cause:** Could not find the password callback class in the classpath. **Action:** Make sure the password callback class to check UsernameToken that decrypts an encrypted message in WSS is in the classpath.
100102008	Unable to convert XML into Document.	Type: SEVERE:NIDP:WSF:008 **Cause:** This error occurred when converting XML to Document in WSS (Receiver side). It may happen due to incorrect WSC requests. **Action:** Check the WSC (Sender side) request and resend it.
100102009	Unable to process WSSecurity (WSS) message.	Type:SEVERE:NIDP:WSF:009 **Cause:** This error occurred when processing WSS headers (Receiver side). It may happen due to incorrect WSS headers in WSC requests. **Action:** Check the WSS headers in WSC (Sender side) request and resent it.
100102010	No WSS header found	Type: SEVERE:NIDP:WSF:010 **Cause:** This error occurred when processing WSS headers (Receiver side). It may happen due to no WSS headers in WSC requests. **Action:** Check the WSS headers in WSC (Sender side) request and resend it.

Event Code	Message	Remedy
100102011	No processed WSS header found	Type: SEVERE:NIDP:WSF:011 **Cause:** This error occurred after processing WSS headers (Receiver side). It may happen due to incorrect or no WSS headers in WSC requests. **Action:** Check the WSS headers in WSC (Sender side) request and resend it.
100102012	WSS untrusted certificate	Type: SEVERE:NIDP:WSF:012 **Cause:** This error occurred when validating signature on WSS headers (Receiver side). The certificate used for the signature is not trusted. **Action:** Check the certificate used to sign the message. The certificate is trusted if either it itself or the certificate of the issuer is installed in the trust store.
100102013		Type: SEVERE:NIDP:WSF:013
100102014		Type: SEVERE:NIDP:WSF:014
100102015		Type: SEVERE:NIDP:WSF:015
100102016		Type: SEVERE:NIDP:WSF:016
100102017		Type: SEVERE:NIDP:WSF:017
100102018		Type: SEVERE:NIDP:WSF:018
100102019		Type: SEVERE:NIDP:WSF:019
100102020		Type: SEVERE:NIDP:WSF:020
100102021		Type: SEVERE:NIDP:WSF:021
100102022		Type: SEVERE:NIDP:WSF:022
100102023		Type: SEVERE:NIDP:WSF:023
100102024		Type: SEVERE:NIDP:WSF:024
100102025	The Service Discovery Service has not been initialized.	Type: SEVERE:NIDP:WSF:025 **Cause:** The Discovery Service has not been enabled or created. **Action:** Create and enable a Liberty Discovery Service using the Access Manager Appliance administration utility.
100102026		Type: SEVERE:NIDP:WSF:026
100102027		Type: SEVERE:NIDP:WSF:027
100102028		Type: SEVERE:NIDP:WSF:028
100102029		Type: SEVERE:NIDP:WSF:029
100102030		Type: SEVERE:NIDP:WSF:030
100102031		Type: SEVERE:NIDP:WSF:031
100102032		Type: SEVERE:NIDP:WSF:032

Event Code	Message	Remedy
100102033		Type: SEVERE:NIDP:WSF:033
100103001	Web Service Consumer XML Configuration Parse Exception.	Type: SEVERE:NIDP:WSC:001 **Cause:** The nidsConfigXML attribute on the nidsWsf object has invalid XML. **Action:** Delete the nidsConfigXML attribute and reconfigure WSC.
100103002		Type: SEVERE:NIDP:WSC:002
100103003		Type: SEVERE:NIDP:WSC:003
100103004		Type: SEVERE:NIDP:WSC:004
100103005		Type: SEVERE:NIDP:WSC:005
100103006		Type: SEVERE:NIDP:WSC:006
100103007		Type: SEVERE:NIDP:WSC:007
100103008		Type: SEVERE:NIDP:WSC:008
100103009		Type: SEVERE:NIDP:WSC:009
100103010		Type: SEVERE:NIDP:WSC:010
100103011		Type: SEVERE:NIDP:WSC:011
100103012		Type: SEVERE:NIDP:WSC:012
100103013		Type: SEVERE:NIDP:WSC:013
100103014		Type: SEVERE:NIDP:WSC:014
100103015		Type: SEVERE:NIDP:WSC:015
100103016		Type: SEVERE:NIDP:WSC:016
100103017		Type: SEVERE:NIDP:WSC:017
100104105	Could not initialize Kerberos/ GSS	Type: SEVERE:NIDP:USERAUTH:105 **Cause:** Failure at GSS-API **Action:** Check the following according the details of the error message: Keytab file - validity, presently only understands DES; Service Principal Name (SPN)
100104107	Kerberos Configuration is not properly initialized	Type: SEVERE:NIDP:USERAUTH:107 **Cause:** Kerberos Configuration is not properly initialized in the admin user interface **Action:** Make sure all the required configuration setting are properly specified in admin UI

Event Code	Message	Remedy
100104108	SPNEGO/Kerberos method not implemented	Type: SEVERE:NIDP:USERAUTH:108 **Cause:** SPNEGO/Kerberos NegTokenInit not implemented. **Action:** NegTokenInit token not implemented as the server side does not need to generate it new. No Action needed.
100105001	An error happened while forwarding a request to a cluster member.	Type: SEVERE:NIDP:APP:001 **Cause:** An internal error occurred. **Action:** Evaluate the error and take appropriate action.
100105002	Failed to initialize JNDI connections.	Type: SEVERE:NIDP:APP:002 **Cause:** NIDP attempts to create JNDI connections to each user store replica during NIDP startup. In this case, NIDP was unable to establish connections with the indicated host. **Action:** Ensure that the host is available and that the configuration information for the replica is correct.
100105003	Error obtaining SOAP response.	Type: SEVERE:NIDP:APP:003 **Cause:** A SOAP request was made and a response was expected, but an error happened retrieving the response. **Action:** Evaluate the indicated reason and take appropriate action.
100105004	Error in SOAP response format.	Type: SEVERE:NIDP:APP:004 **Cause:** A SOAP request was made and a response was expected, the response was obtained but the format of it was unexpected. **Action:** Evaluate the indicated reason and take appropriate action.
100105005	Error executing Login Policy Check LDAP Extension for user on user store	Type: SEVERE:NIDP:APP:005 **Cause:** User authenticated using X509. An additional check of the directory's user login policy needs to be made using an LDAP method extension. This check was successfully done using an LDAP extension. However, after the LDAP extension is called, it must be called a second time to update the user account with a success or failure. This second call to the extension failed, so directory user account status may be erroneous. **Action:** Check with eDirectory documentation for LDAP extension with OID 2.16.840.1.113719.1.39.42.100.25
100105006		Type: SEVERE:NIDP:APP:006
100105007		Type: SEVERE:NIDP:APP:007

Event Code	Message	Remedy
100105008	The audit logging system is not operational.	Type: SEVERE:NIDP:APP:008 **Cause:** The audit logging system can, in rare circumstances, become non-operational. **Action:** Examine the error description supplied and take appropriate action.
100106001		Type: SEVERE:NIDP:IDFF:001
200102001	Invalid access code found for web service specific user interaction query policy.	Type: ERROR:NIDP:WSF:001 **Cause:** The web service definition has a service level user interaction policy that is not ALWAYS or NEVER. Disallowed values are NO and ONCE. **Action:** Using Access Manager Appliance management tools, edit the policy associated with the web service.
200102002	Invalid access code found for web service specific user interaction modify policy.	Type: ERROR:NIDP:WSF:002 **Cause:** The web service definition has a service level user interaction policy that is not ALWAYS or NEVER. Disallowed values are NO and ONCE. **Action:** Using Access Manager Appliance management tools, edit the policy associated with the web service.
200102003	Unrecognized web service.	Type: ERROR:NIDP:WSF:003 **Cause:** The web service definition has a service type specifier (attribute nidsWsfServiceInstanceType on object nidsWsfService) that is not recognized. **Action:** Using Access Manager Appliance management tools, delete the associated web service and recreate it.
200102004	Error writing user interaction access policy to the data store.	Type: ERROR:NIDP:WSF:004 **Cause:** The IDP received user interaction access policy from the user, but was unable to persist it to the data store. **Action:** Check the Access Manager Appliance Configuration datastore to see if it is available.
200102005	Cannot read or write web service data because zero data locations are specified.	Type: ERROR:NIDP:WSF:005 **Cause:** When an IDSIS web service is reading or writing data it follows the configured data locations to know where to perform its operations. If the administrator has not set up any data locations then the operation must fail. **Action:** Add at least one data location the web service.

Event Code	Message	Remedy
200102006	Cannot read or write web service data because the first data location is unknown.	Type: ERROR:NIDP:WSF:006 **Cause:** When an IDSIS web service is reading or writing data it follows the configured data locations to know where to perform its operations. **Action:** Delete all data locations from the associated web service and add them back into the list.
200102007	Unexpected error writing data to web service.	Type: ERROR:NIDP:WSF:007 **Cause:** Writing to web services is prone to various unexpected errors. **Action:** Evaluate the reason for the error and take appropriate action.
200102008	Unable to locate the cached NIDPSession object given session id.	Type: ERROR:NIDP:WSF:008 **Cause:** The user session has expired. **Action:** The user must login again.
200102009	Cached NIDPPrincipal object has zero NIDPSubject objects.	Type: ERROR:NIDP:WSF:009 **Cause:** The user session has expired. **Action:** The user must login again.
200102010	No web service authority available.	Type: ERROR:NIDP:WSF:010 **Cause:** A web service of the provided type did not initialize correctly. **Action:** Delete the web service and recreate it.
200102011	No web service available.	Type: ERROR:NIDP:WSF:011 **Cause:** A web service of the provided type does not exist, or is not enabled. **Action:** Create or enable a web service of this type.
200102012	Unable to understand the web service request's XML.	Type: ERROR:NIDP:WSF:012 **Cause:** A web service sent a request to the IDP that cannot be parsed or it is missing data such that the request cannot be understood. **Action:** Notify your system administrator that invalid web service requests are being made to the system.
200102013	Error processing web service query request.	Type: ERROR:NIDP:WSF:013 **Cause:** Processing web service requests may result in a number of unexpected errors. **Action:** Evaluate the reason given in the error message, and take appropriate action.

Event Code	Message	Remedy
200102014	Error processing web service modify request.	Type: ERROR:NIDP:WSF:014 **Cause:** Processing web service requests may result in a number of unexpected errors. **Action:** Evaluate the reason given in the error message, and take appropriate action.
200102015	Unable to locate the user's local identifier in the resource id.	Type: ERROR:NIDP:WSF:015 **Cause:** The web service resource id, an identifier indicating what user the request is destined for, did not contain the information required to identify the user. **Action:** Notify your system administrator that invalid web service requests are being made to the system.
200102016	Unable to locate a cached NIDPPrincipal object given the local id.	Type: ERROR:NIDP:WSF:016 **Cause:** The user session has expired. **Action:** The user must login again.
200102017	Unable to locate a NIDPIdentity object given the local id.	Type: ERROR:NIDP:WSF:017 **Cause:** The user session has expired. **Action:** The user must login again.
200103001	The indicated web service is not available or it has been disabled! An attempt was made to access this service to operate on the indicated data.	Type: ERROR:NIDP:WSC:001 **Cause:** The Web Service Consumer received a request and one of the data tokens referenced a data item that is not available in any of the services known to the Access Manager Appliance. **Action:** The system has encountered an invalid configuration and should be restarted by the system administrator.
200103002	Cannot make web service request because there are zero web service resource offerings available.	Type: ERROR:NIDP:WSC:002 **Cause:** The Web Service Consumer received a request but there were zero service resource offerings provided. So, the web service has no destination service to which a request can be made. **Action:** The user must login again.
200103003	Unable to locate an identity id from the authentications available in the provided NIDPSession.	Type: ERROR:NIDP:WSC:003 **Cause:** The user session has expired. **Action:** The user must login again.
200104001	Could not get client certificate.	Type: ERROR:NIDP:USERAUTH:001 **Cause:** Could not get user certificate from the client browser **Action:** Install user X509 certificate on the client browser and try again.

Event Code	Message	Remedy
200104003	Could not read configuration	Type: ERROR:NIDP:USERAUTH:003 **Cause:** Could not read configuration out of file **Action:** Make sure the X509 config properties file is present.
200104004	User Certificate Authentication Failed	Type: ERROR:NIDP:USERAUTH:004 **Cause:** User Certificate Authentication Failed due to the reasons in detailed message **Action:** Take appropriate action as per the reasons in the detailed message
200104005	No matching Principal found.	Type: ERROR:NIDP:USERAUTH:005 **Cause:** No Principal from X509Certificate found in User store **Action:** Check the X509Class Method and it's attribute mapping profile as defined using administration tool. Also, make sure the matched user exists in the User store.
200104006	More than one Principal matched.	Type: ERROR:NIDP:USERAUTH:006 **Cause:** Principal from X509Certificate Multiple users found in User store which matched Principal from X509Certificate based on X509Class attribute mapping profile.\ **Action:** Check the X509Class Method and it's attribute mapping profile as defined using administrator tool. Also, check if multiple user exists in the User store(s).
200104008	Error loading Trust store	Type: ERROR:NIDP:USERAUTH:008
200104009	Client certificate not yet valid.	Type: ERROR:NIDP:USERAUTH:009 **Cause:** X509 certificate is valid in the future **Action:** Use a valid certificate
200104010	Client certificate no longer valid.	Type: ERROR:NIDP:USERAUTH:010 **Cause:** X509 certificate is expired **Action:** Use a valid certificate
200104011	The Certificate has been revoked.	Type: ERROR:NIDP:USERAUTH:011 **Cause:** The Certificate has been revoked **Action:** Use a valid certificate which is not revoked.
200104012	Error Parsing Certificate.	Type: ERROR:NIDP:USERAUTH:012 **Cause:** Error Parsing Certificate when performing certificate validations **Action:** Use a valid X509 certificate.

Event Code	Message	Remedy
200104017	Error getting CRL/OCSP.	Type: ERROR:NIDP:USERAUTH:017 **Cause:** Could not get to the CRL/OCSP URL for validations. **Action:** Make sure the CRL/OCSP URLs are accessible Or disable validations in administration. Additionally, can define a different CRL/OCSP URL in the administration tool which the X509Class can also use for validations.
200104018	Could not verify CRL signature.	Type: ERROR:NIDP:USERAUTH:018 **Cause:** Could not verify signature on the fetched CRL **Action:** Make sure the CRL server public key/certificate is in NIDP/ESP trust store.
200104019	Could not find Key for this server.	Type: ERROR:NIDP:USERAUTH:019 **Cause:** Could not find Key/Cert for NIDP/ESP server towards authenticating to OCSP server **Action:** Make sure the NIDP/ESP Signing keystore has appropriate Key/Cert in it.
200104020	CRL/OCSP is too old; New version already available.	Type: ERROR:NIDP:USERAUTH:020 **Cause:** During validations, the fetched CRL Or OCSP is stale. Newer version will be available **Action:** In case of CRLs, next attempt to fetch CRL should get a fresh CRL after purging the cached one. In case of OCSP, notify the OCSP server administrator.
200104021	No Issuer Certificate found.	Type: ERROR:NIDP:USERAUTH:021 **Cause:** Issuer of user certificate not found which is required for OCSP validations **Action:** Make sure the issuer of user/client certificate is either found in certificate-chain or in NIDP/ESP trust store.
200104022	Error getting OCSP Response.	Type: ERROR:NIDP:USERAUTH:022 **Cause:** Could not get OCSP Response from the OCSP server **Action:** Make sure its going to the right OCSP server.
200104023	Error processing OCSP Response.	Type: ERROR:NIDP:USERAUTH:023 **Cause:** OCSP response could not be processed **Action:** Make sure its going to the right OCSP server and that it is operating correctly.

Event Code	Message	Remedy
200104024	At least one parameter of OCSPProcessor was uninitialized.	Type: ERROR:NIDP:USERAUTH:024 **Cause:** At least one parameter of OCSPProcessor was uninitialized during OCSP validations **Action:** Make sure the NIDP/ESP Signing keystore has appropriate Key/Cert in it. Also, that the NIDP/ESP OCSP trust store has the valid public-key/certificate of OCSP server.
200104025	Request was already generated.	Type: ERROR:NIDP:USERAUTH:025 **Cause:** OCSP request was already generated for certificate(s) **Action:** Check the client certificate chain.
200104026	OCSP response was already processed	Type: ERROR:NIDP:USERAUTH:026
200104027	Internal error occurred in the OCSP Server.	Type: ERROR:NIDP:USERAUTH:027 **Cause:** OCSP server responded to the request with an internal error. **Action:** Contact OCSP server administrator.
200104028	Your request did not fit the RFC 2560 syntax.	Type: ERROR:NIDP:USERAUTH:028 **Cause:** OCSP server responded to the request with malformed request message. **Action:** Contact OCSP administrator and check the request.
200104029	Your request was not signed.	Type: ERROR:NIDP:USERAUTH:029 **Cause:** Request to OCSP server needs to be signed. **Action:** Enable signing of OCSP requests in X509Class administration.
200104030	The server was too busy to answer you.	Type: ERROR:NIDP:USERAUTH:030 **Cause:** OCSP server is too busy to respond to requests. **Action:** Contact OCSP server administrator.
200104031	The server could not authenticate you.	Type: ERROR:NIDP:USERAUTH:031 **Cause:** OCSP server could not authenticate Novell Identity server. **Action:** Make sure Signing of OCSP requests is enabled and NIDP signing keystore has appropriate key in it. Also, make sure the OCSP server trusts Nidp server.

Event Code	Message	Remedy
200104032	Unknown OCSPResponse status code.	Type: ERROR:NIDP:USERAUTH:032 **Cause:** OCSP server responded to the request with unknown status code. **Action:** Contact OCSP server administrator.
200104033	No valid OCSPResponse obtained.	Type: ERROR:NIDP:USERAUTH:033 **Cause:** Invalid OCSP response obtained. **Action:** Check the OCSP server response version and contact administrator.
200104034	Response was generated in the future.	Type: ERROR:NIDP:USERAUTH:034 **Cause:** OCSP response is not yet valid. **Action:** Disable OCSP validations Or Contact OCSP server administrator.
200104035	Error verifying responder certificate.	Type: ERROR:NIDP:USERAUTH:035 **Cause:** This may happen when reading the OCSP trust store during OCSP validations. **Action:** Make sure OCSP trust store exists on NIDP server.
200104036	Response seems to be signed with untrusted certificate.	Type: ERROR:NIDP:USERAUTH:036 **Cause:** OCSP server trusted-root certificate not found in OCSP trust store. **Action:** Import OCSP server trusted root in Nidp's OCSP trust store.
200104037	The received responder id does not match your responder certificate.	Type: ERROR:NIDP:USERAUTH:037 **Cause:** The response ID received in OCSP response does not match. **Action:** Make sure NIDP's OCSP trust store has the right OCSP server public-key certificate.
200104038	Could not verify OCSP server response.	Type: ERROR:NIDP:USERAUTH:038 **Cause:** OCSP server response is incorrect. **Action:** Verify the OCSP server URL. Make sure NIDP's OCSP trust store has the right OCSP server public-key certificate.
200104039	No client certificates inside OCSP response.	Type: ERROR:NIDP:USERAUTH:039 **Cause:** Empty response from OCSP server. **Action:** Verify the OCSP server URL.

Event Code	Message	Remedy
200104040	Number of certificates inside OCSP response does not fit to request.	Type: ERROR:NIDP:USERAUTH:040 **Cause:** OCSP response does not contain the requested number of certificate status. **Action:** Verify the OCSP server URL.
200104041	Certificate was revoked in the future.	Type: ERROR:NIDP:USERAUTH:041 **Cause:** OCSP response not yet valid. **Action:** Verify the OCSP server URL.
200104042	Received certificate twice or one, that was not requested.	Type: ERROR:NIDP:USERAUTH:042 **Cause:** OCSP response does not match request. **Action:** Verify the OCSP server URL.
200104043	Request was not accepted.	Type: ERROR:NIDP:USERAUTH:043 **Cause:** Could not connect to OCSP server. **Action:** Verify the OCSP server URL.
200104044	Wrong response type (not application/ocsp-response).	Type: ERROR:NIDP:USERAUTH:044 **Cause:** Malformed OCSP response. **Action:** Verify the OCSP server URL.
200104045	No OCSPResponse message.	Type: ERROR:NIDP:USERAUTH:045 **Cause:** No OCSPResponse message. **Action:** Verify the OCSP server URL.
200104046	Could not read whole OCSPResponse.	Type: ERROR:NIDP:USERAUTH:046 **Cause:** Malformed OCSP response. **Action:** Verify the connection to OCSP server URL.
200104047	Exception Occurred.	Type: ERROR:NIDP:USERAUTH:047 **Cause:** Error getting CRL. **Action:** Verify the connection to CRL server URL.
200104051	Unsupported critical extension OID(s).	Type: ERROR:NIDP:USERAUTH:051 **Cause:** Some Critical extension OID(s) not understood. **Action:** Check the certificate for unsupported critical extensions. If needed, add the processing of the critical extension in NDPCertPathChecker class.
200104053	Error processing CRL Response.	Type: ERROR:NIDP:USERAUTH:053 **Cause:** Error processing CRL Response. **Action:** Check X509class config and user/client certificate CRL extension.

Event Code	Message	Remedy
200104054	Error processing certificate validations.	Type: ERROR:NIDP:USERAUTH:054 **Cause:** Error processing CRL/OCSP validations. **Action:** Check X509class config and user/client certificate CRL extension.
200104055	Protocol not supported or none specified.	Type: ERROR:NIDP:USERAUTH:055 **Cause:** Transport protocol not supported to fetch CRL. **Action:** Currently, CRLs can be fetched over http and LDAP protocols. Make sure the X509class config and/or user/client certificate CRL extension does not have any other transport protocol specified.
200104057	Unable to do X509 Certificate based authentication over non SSL (HTTP).	Type: ERROR:NIDP:USERAUTH:057 **Cause:** URL protocol is HTTP **Action:** URL protocol needs to be HTTPS
200104058	Overwrite a real or temp user error.	Type: ERROR:NIDP:USERAUTH:058 **Cause:** User is not identified in the authenticated user session. **Action:** Authenticate with a valid authentication contract to identify the user.
200104059	User store connection error.	Type: ERROR:NIDP:USERAUTH:059 **Cause:** LDAP replica connection error **Action:** Check the connectivity from the Identity server to LDAP replicas.
200104060	Problem in fetching password.	Type: ERROR:NIDP:USERAUTH:060 **Cause:** Error while fetching user password **Action:** Check the password policy for the user and verify that admin has permission to retrieve the password for that user.
200104061	Problem in provisioning the user.	Type: ERROR:NIDP:USERAUTH:061 **Cause:** Error while auto User provisioning for password fetch class. **Action:** Check whether admin has permission to create user and modify user's attributes in the LDAP store.
200104062	Auto provisioning successful.	Type: INFO:NIDP:USERAUTH:062 **Scenario:** Password fetch class was successful in auto provisioning user.

Event Code	Message	Remedy
200104063	Universal password retrieval error.	Type: ERROR:NIDP:USERAUTH:063 **Cause:** Universal password retrieval error with password fetch class. **Action:** Check the universal password policy for the user and verify that admin has permission to retrieve the password for that user.
200104064	Simple password retrieval error.	Type: ERROR:NIDP:USERAUTH:064 **Cause:** Simple password retrieval error with password fetch class. **Action:** Check the simple password policy for the user and verify that admin has permission to retrieve the password for that user.
200104065	User lookup failed.	Type: ERROR:NIDP:USERAUTH:065 **Cause:** User lookup failed for the Distinguished Name (DN) with password fetch class. **Action:** Create a user DN in the eDirectory from which the user password is retrieved.
200104066	Client Integrity Check (CIC) failed.	Type: ERROR:NIDP:USERAUTH:066 **Cause:** CIC failed. **Action:** Check if the required software is available on the system. Check the CIC class is configured properly. Check the logs on java console on the system.
200104100	Error processing Authorization header	Type: ERROR:NIDP:USERAUTH:100 **Cause:** Could not process HTTP Authorization header **Action:** Try with correct authorization header with base64 encoded SPNEGO token
200104101	Error processing SPNEGO/ Kerberos	Type: ERROR:NIDP:USERAUTH:101 **Cause:** Error processing SPNEGO/Kerberos. The cause is included in detailed message **Action:** Take action as per the detailed error message
200104102	No Kerberos Principal found in the token	Type: ERROR:NIDP:USERAUTH:102 **Cause:** Failure at GSS-API **Action:** Make sure the Kerberos keytab file is generated correctly by KDC
200104103	No SPNEGO Token found	Type: ERROR:NIDP:USERAUTH:103 **Cause:** No SPNEGO Token found in the request **Action:** Include the SPNEGO token in the request to use this authentication

Event Code	Message	Remedy
200104104	GSS Context already established	Type: ERROR:NIDP:USERAUTH:104 **Cause:** GSS Context already established **Action:** Close the browser and try again
200104106	Unrecognized SPNEGO Token	Type: ERROR:NIDP:USERAUTH:106 **Cause:** Unrecognized SPNEGO Token **Action:** Include the correct SPNEGO token in the request to use this authentication
200104109	Malformed SPNEGO NegTokenInit	Type: ERROR:NIDP:USERAUTH:109 **Cause:** Malformed token NegTokenInit **Action:** Try again with correct NegTokenInit token
200104110	Malformed SPNEGO Token field	Type: ERROR:NIDP:USERAUTH:110 **Cause:** Malformed SPNEGO Token field **Action:** Try again with correct NegTokenInit token
200104111	Multiple users matched in the user stores	Type: ERROR:NIDP:USERAUTH:111 **Cause:** Multiple users matched in the user stores **Action:** Make sure the users are unique in user stores
200104112	No user matched in the user stores	Type: ERROR:NIDP:USERAUTH:112 **Cause:** No user found in the user stores **Action:** Make sure the user attribute (as defined in admin UI) is populated in correct format.
200107005	Error building certificate chain during validations.	Type: ERROR:NIDP::005 **Cause:** This could occur when all the CDPs are unreachable. **Action:** Change the Certificate with correct CDPs or make sure CDP is up and able to serve.
300101002	An authenticated subject is required.	Type: WARN:NIDP:USERMSG:002 **Cause:** An action that can only be performed by an authenticated user was attempted. **Action:** Provide proper user credentials and retry desired action.
300101003	An authentication principal is required.	Type: WARN:NIDP:USERMSG:003 **Cause:** An action that can only be performed by an authenticated user was attempted. **Action:** User must be authenticated to perform operation.

Event Code	Message	Remedy
300101004	Identity does not exist or is not specified.	Type: WARN:NIDP:USERMSG:004 **Cause:** An action was attempted that requires a federated identity to exist. **Action:** Create a federated link prior to performing the action.
300101005	Invalid or no provider is specified.	Type: WARN:NIDP:USERMSG:005 **Cause:** An action was requested related to a trusted provider that does not exist. **Action:** Add the desired provider as a trusted entity or check for invalid access to system.
300101006	An authenticated session is required.	Type: WARN:NIDP:USERMSG:006 **Cause:** An action that can only be performed by an authenticated user was attempted. **Action:** Provide proper user credentials and retry desired action.
300101007	Invalid artifact.	Type: WARN:NIDP:USERMSG:007 **Cause:** An artifact was received from an identity provider that is invalid or has not been used within a reasonable time frame. **Action:** Make sure that the provider sending the artifact is trusted or check for possible security intrusions.

Event Code	Message	Remedy
300101008	No assertion returned in response. No authentication context specified message in the assertion.	Type: WARN:NIDP:USERMSG:008 **Cause:** Assertions will not be returned in a response whenever authentication at the identity provider fails. The cause for this can include invalid configurations and canceling the authentication process at the identity provider. This response is also returned when a user has reached the maximum number of sessions and then attempts to access a protected resource that requires authentication. **Action:** Make sure that both the identity and service providers are configured correctly to trust each other. Provide proper credentials during the authentication process at the identity provider. **Cause:** Protected resources are configured to access using external contracts, which are being executed at the external identity provider. These contracts are not configured to be satisfied by any of the external identity provider. **Action1:** Verify the external identity provider satisfiable contract list at the service provider and ensure that these external contracts are configured under the satisfiable list. **Action 2:** Verify the external contract definition at the identity provider and make sure that this contract definition with the matching allowable class or URI is available. **NOTE:** URI specifies a value that uniquely identifies the contract from all other contracts.
300101009	Invalid issuer.	Type: WARN:NIDP:USERMSG:009 **Cause:** A response was received from a provider that is not trusted. **Action:** Make sure intended provider is trusted or check for possible intrusions.
300101010	Response does not match request.	Type: WARN:NIDP:USERMSG:010 **Cause:** A response was received for a request that was not issued. **Action:** Retry action and check for possible intrusion.
300101011	Assertion is being replayed.	Type: WARN:NIDP:USERMSG:011 **Cause:** An assertion has been received that was already used to authenticate a user at the service provider. **Action:** This is a security mechanism that if persists may require some investigation to determine who is trying to replay the assertion. Assertions are only good for single use.

Event Code	Message	Remedy
300101012	Assertion does not contain an authentication statement.	Type: WARN:NIDP:USERMSG:012 **Cause:** An identity provider has sent an assertion that is not complete. **Action:** Check with administrator of trusted provider to determine why statement is not being sent.
300101013	Unable to validate the subject of the assertion.	Type: WARN:NIDP:USERMSG:013 **Cause:** A subject may not have been sent in the assertion or was not valid. This check protects from certain assertion attacks. If the time is not in sync between the identity provider and the service provider, the subject is invalid because of the timestamp sent with the subject. **Action:** If persistent, check the protocol message sent for a time discrepancy between the providers or a missing subject, then notify the administrator of the trusted site. For more information, see "Federation with External SAML 2.0 Partner Gives 300101013 Error" (http://www.novell.com/support/php/search.do?cmd=displayKC&docType=kc&externalId=3903427&sliceId=2&docTypeID=DT_TID_1_1&dialogID=69860557&stateId=0%200%2069862016).
300101014	Assertion not yet valid.	Type: WARN:NIDP:USERMSG:014 **Cause:** An assertion was received that is not valid until sometime in the future. **Action:** Check server's clock for accuracy. Attempt to validate the clock accuracy of the computer generating the assertion.
300101015	Assertion no longer valid.	Type: WARN:NIDP:USERMSG:015 **Cause:** An assertion was received that had a time validity period that is in the past. **Action:** Check server's clock for accuracy. Attempt to validate the clock accuracy of the computer generating the assertion. Try to authenticate again.
300101016	No matching audience.	Type: WARN:NIDP:USERMSG:016 **Cause:** An assertion was received that was not intended for your server. **Action:** Determine the origin of the assertion and make sure that you want to accept assertions from it. For more information, see "Access Manager 300101016 Error - No Matching Audience" (http://www.novell.com/support/php/search.do?cmd=displayKC&docType=kc&externalId=3260366&sliceId=2&docTypeID=DT_TID_1_1&dialogID=69860436&stateId=0%200%2069856899).

Event Code	Message	Remedy
300101017	Missing or invalid signature on assertion.	Type: WARN:NIDP:USERMSG:017 **Cause:** The identity provider did not sign. **Action:** Check with provider of assertion to determine why assertion is not signed.
300101018	Missing or invalid signature on request/response.	Type: WARN:NIDP:USERMSG:018
300101020	Digital signature is required.	Type: WARN:NIDP:USERMSG:020 **Cause:** A protocol message was received that was expected to be digitally signed, but was not. **Action:** It may be necessary to contact the trusted provider administrator to determine why the message is not signed. Make sure authentication request signing settings match those for the trusted provider.
300101021	Signature validation failed.	Type: WARN:NIDP:USERMSG:021 **Cause:** The digital signature of a protocol message could not be verified using the public key obtained in the metadata of a trusted provider. **Action:** Update the metadata of trusted provider. This should ensure you have the latest signing certificate.
300101022	An undetermined problem in the message format has occurred.	Type: WARN:NIDP:USERMSG:022 **Cause:** An error was detected in the exchange of either a Liberty or SAML protocol message. **Action:** Turn logging/tracing on to print out the message that is problematic. It may be necessary to contact Novell Technical Services in this case.
300101023	User lookup failed.	Type: WARN:NIDP:USERMSG:023 **Cause:** An attempt to identify a user failed while attempting to complete a federation at the server. **Action:** Check the configuration for identifying users for the trusted provider and ensure the specified method can resolve to a single user in your directory.
300101024	Failed to load java class.	Type: WARN:NIDP:USERMSG:024 **Cause:** A Java class failed to be loaded during program execution. **Action:** Check the logs to determine the class that is failing to load. Make sure the class being loaded is in the classpath of the JVM.
300101025		Type: WARN:NIDP:USERMSG:025
300101026		Type: WARN:NIDP:USERMSG:026
300101027		Type: WARN:NIDP:USERMSG:027

Event Code	Message	Remedy
300101028	SOAP TLS authorization failed.	Type: WARN:NIDP:USERMSG:028 **Cause:** SSL mutual authentication is being used to authenticate a SOAP back channel session and the credentials cannot be validated. **Action:** Make sure certificates for back channel communications are trusted on each end. For more information, see "Access Manager 300101028 - SOAP TLS Authorization Failed" (http://www.novell.com/support/php/search.do?cmd=displayKC&docType=kc&externalId=3813149&sliceId=2&docTypeID=DT_TID_1_1&dialogID=69848431&stateId=0%200%2069844751).
300101029		Type: WARN:NIDP:USERMSG:029
300101030	SOAP fault.	Type: WARN:NIDP:USERMSG:030 **Cause:** An error was detected in the transmission of protocols using SOAP. **Action:** Turn tracing on and look for any obvious causes for the problem.
300101031	Received an identity that does not resolve to the current logged in user.	Type: WARN:NIDP:USERMSG:031 **Cause:** This is caused when a user is logged in with one identity and then attempts to authenticate as the identity of another user. For a given session, all authentications must resolve to the same user. **Action:** Log out of the current user and log in again as the desired user.
300101032	Assertion is expired.	Type: WARN:NIDP:USERMSG:032 **Cause:** The use of the assertion to authenticate the server did not occur within the time limits specified by the assertion. **Action:** Try and re-authenticate. Determine if there are any network latencies that may cause the assertion not to arrive in a timely fashion. Look for misuse of the assertion.
300101033	IDP return authentication failure.	Type: WARN:NIDP:USERMSG:033 **Cause:** An IDP's attempt to authenticate the server was unsuccessful. This particular authentication came from the IDP's intersite transfer service and was not requested by the server. **Action:** Check at the IDP for a reason why the authentication was a failure. It may just be necessary to attempt authentication again.

Event Code	Message	Remedy
300101034	No target is defined.	Type: WARN:NIDP:USERMSG:034 **Cause:** A request was made of the server's intersite transfer service without specifying a target resource. **Action:** Requests for the intersite transfer service must include an id of the intended service provider to be authenticated as well as the target resource to be displayed. To avoid this error, provide an &TARGET="value" on the URL.
300101035		Type: WARN:NIDP:USERMSG:035
300101036	Not enough memory to process request.	Type: WARN:NIDP:USERMSG:036 **Cause:** The system does not have enough memory to complete the requested action. **Action:** Wait a few moments for memory to free up and retry request. It may be necessary to add additional memory to the server.
300101037	Server is not in a running state.	Type: WARN:NIDP:USERMSG:037 **Cause:** A request was made of the server that can only be performed when the server is in a running state. **Action:** Start the server.
300101038	JSP file not found.	Type: WARN:NIDP:USERMSG:038 **Cause:** An attempt was made to load a JSP page that does not exist. **Action:** Determine the JSP not loading and make sure it is in the correct location.
300101039	Invalid authentication credentials were provided.	Type: WARN:NIDP:USERMSG:039 **Cause:** A user has attempted to authenticate to the system with credentials that are not valid for the account. **Action:** User needs to enter correct credentials.
300101040	User password has expired.	Type: WARN:NIDP:USERMSG:040 **Cause:** A user has attempted to authenticate to the system with a password that is expired. **Action:** The user needs to create a new password.

Event Code	Message	Remedy
300101041	User account identification failed.	Type: WARN:NIDP:USERMSG:041 **Cause:** Account identification can fail due to: 1. User cancels authentication request 2. User cannot be uniquely identified by Matching Expression 3. Necessary attributes to do user matching or provisioning were not obtained. **Action:** Check Account Identification configuration for the trusted provider and make sure that necessary attributes are available. If using Matching Expressions, make sure that they include attributes that can resolve to a single user. If using Provisioning, make sure required attributes are all available in the defined attribute set for the trusted provider. For more information, see "Access Manager Error 300101041 Provisioning New Users Using SAML2" (http://www.novell.com/support/php/search.do?cmd=displayKC&docType=kc&externalId=3219302&sliceId=1&docTypeID=DT_TID_1_1&dialogID=69780245&stateId=0%200%2069778277).
300101042	Invalid assertion conditions.	Type: WARN:NIDP:USERMSG:042 **Cause:** A set of conditions that are not understood were sent as part of an assertion. **Action:** Check with the provider of the assertion to determine what these conditions are and why they are being sent.
300101046	Unknown URL host.	Type: WARN:NIDP:USERMSG:046 **Action:** Use logs to determine the problematic host and determine why DNS is failing.
300101047	An untrusted provider is being referenced in a request or a response.	Type: WARN:NIDP:USERMSG:047 **Action:** Use logs to determine the provider that is untrusted and then create a trusted relationship if desired.
300101048	The LDAP servers are too busy to accept more users.	Type: WARN:NIDP:USERMSG:048 **Cause:** There are too many threads waiting to get an available LDAP connection. The LDAP servers are too busy to accept more users. **Action:** Wait a few moments for the LDAP requests to be processed and retry the request. It may be necessary to add additional LDAP servers or upgrade the hardware specifications of the existing LDAP servers.
300101049	The HTTPS protocol was not used to access this authentication card.	Type: WARN:NIDP:USERMSG:049 **Cause:** Accessing the site was done via http, not https. **Action:** Access the site again using https.

Event Code	Message	Remedy
300101050	The Authentication Card specified is not valid.	Type: WARN:NIDP:USERMSG:050 **Cause:** An invalid card identifier was used, most likely due to modifying a url. **Action:** Specify cards to use only by clicking them. Ensure that the PID in the login URL exactly matches the entity ID specified in the metadata.
300101051	The user's session limit has been reached.	Type: WARN:NIDP:USERMSG:051 **Cause:** User has already logged in the maximum allowable times. **Action:** Logout of one or more sessions.
300101052	A response was expected at the url but none was found.	Type: WARN:NIDP:USERMSG:052 **Cause:** The wrong endpoint may be accessed for the operation desired. **Action:** Check the action being performed against the url/endpoint being accessed.
300101053	CardSpace authentication profile failed to load.	Type: WARN:NIDP:USERMSG:053 **Cause:** Trusted provider failed to load (probably due to certificate errors). **Action:** Check the certificates for the trusted provider and make sure they are valid.
300101054	CardSpace authentication fails because a required attribute is not in assertion.	Type: WARN:NIDP:USERMSG:054 **Cause:** A required attribute was not returned in the assertion provided by an STS. **Action:** Check the attribute value at the STS, or make the attribute optional.
300101057	Request to broker an authentication to a target SP denied	Type: WARN:NIDP:USERMSG:057 **Cause:** Request to broker an authentication to a target Service Provider denied, either the Identity Provider or target Service Provider are part of a brokering group, but both does not belong to same group. **Action:** Check the brokering group to verify if the Identity Provider and target Service Provider belong to the same group.
300101058	Request to broker on authentication to a target SP denied	Type: WARN:NIDP:USERMSG:058 **Cause:** Request to broker on authentication to a target Service Provider denied because broker policy evaluation resulted in denying. **Action:** Check the brokering rule and try to access with the valid user.

Event Code	Message	Remedy
300101059	Error in processing the broker request.	Type: WARN:NIDP:USERMSG:059 **Cause:** Could not validate the request to broker on authentication to a target service provider. **Action:** Read the error description supplied and take appropriate action.
300101060	Assertion does not contain valid authentication statement.	Type: WARN:NIDP:USERMSG:060 **Cause:** An assertion has been received, which does not contain valid declaration/class statement. **Action:** Authentication response statement will be validated against the authentication request statement. Check the contract definition in the service provider for the authentication statement received from the Identity Provider. Check if the requested statement matches the response statement or response statement's authentication level is greater than the requested one.
300101061	Failed to obtain consent for the federation.	Type: ERROR:NIDP:USERMSG:061 **Cause:** This is a message to users if they have declined the consent. If it is a valid federation consent, accept the consent in the next attempt, else deny the same. **Action:** In a federated setup using the name identifier as persistent, if you make an Intersite Transfer Service request for the first time federation, users will be asked to provide their consent and they select No.
300101062	An Identity Provider response was received that failed to authenticate this session.	**Cause:** When you configure a policy for a spsend request to SAML 2.0, the user is denied the policy rule, and a message is displayed. **Action:** You are accessing an URL that is not intended for you. Contact your administrator.
300102001	No Discovery Service Configured! Unable to create the requested resource offering!	Type: WARN:NIDP:WSF:001 **Cause:** The system administrator did not create or enable a Discovery service. **Action:** Create or enable a Discovery web service.
300102002	Unable to find user object with identifier.	Type: WARN:NIDP:WSF:002 **Cause:** An LDAP search was performed for a user object with a given identifier. This identifier may be a GUID. The search resulted in zero hits. This usually means that web service data cannot be read or written for the user. **Action:** The user needs to login again.

Event Code	Message	Remedy
300102003	Unrecognized select string for service.	Type: WARN:NIDP:WSF:003 **Cause:** The select string (XPath) is either incorrectly formed or not supported by the web service. **Action:** The system administrator must enable services to support the select string.
300102004	Unable to process web service query request! Select string missing!	Type: WARN:NIDP:WSF:004 **Cause:** The select string (XPath) is not in the web service query request. **Action:** Inform your system administrator that an improperly formatted web service request is being made.
300102005	Unable to perform trusted user interaction service request. Web service authority was not found.	Type: WARN:NIDP:WSF:005 **Cause:** An internal system error. **Action:** The system has encountered an invalid configuration and should be restarted by the system administrator.
300102006	Unable to perform trusted user interaction service request. Unable to obtain trusted user interaction service description from SOAP headers.	Type: WARN:NIDP:WSF:006 **Cause:** The web service making the request did not provide valid or complete information about the trusted user interaction service. **Action:** The system administrator must complete the definition of the trusted interaction service.
300102007	Unable to perform trusted user interaction service request. No trusted user interaction service description provided in SOAP headers.	Type: WARN:NIDP:WSF:007 **Cause:** The web service making the request did not provide valid or complete information about the trusted user interaction service. **Action:** The system administrator must complete the definition of the trusted interaction service.
300102008	Trusted user interaction service failed.	Type: WARN:NIDP:WSF:008 **Cause:** There are various unexpected reasons for the failure of a trusted user interaction service request to fail. **Action:** Evaluate the reason and take the appropriate actions.
300102009	Error creating user interaction redirection request.	Type: WARN:NIDP:WSF:009 **Cause:** There was an error converting the redirect request to an XML DOM. **Action:** Evaluate the reason and take the appropriate actions.

Event Code	Message	Remedy
300102010	Unable to perform user interaction redirection request. User intervention service not found.	Type: WARN:NIDP:WSF:010 **Cause:** There must be an interaction service on the IDP creating the user interaction redirection request. **Action:** If it does not exist, using Access Manager Appliance management tools, create one.
300102011	Error reading data from LDAP data attribute plugin.	Type: WARN:NIDP:WSF:011 **Cause:** If a web service's data locations includes LDAP, then LDAP data attribute plugins are used to read data from the LDAP user store. This error provides descriptions of various errors that can happen while doing this. **Action:** Evaluate the reason and take the appropriate actions.
300102012	Error writing data to LDAP data attribute plugin.	Type: WARN:NIDP:WSF:012 **Cause:** If a web service's data locations includes LDAP, then LDAP data attribute plugins are used to write data to the LDAP user store. This error provides descriptions of various errors that can happen while doing this. **Action:** Evaluate the reason and take the appropriate actions.
300102013	Cannot read/write Credential Profile data because the user's LDAP user store distinguished name is not available.	Type: WARN:NIDP:WSF:013 **Cause:** All Credential Profile reads and writes end up operating on a user object in a user store. If this user object cannot be found, then the operation must fail. This may happen if a temporary identifier is being used for the authentication. **Action:** Use a permanent federation to the service provider if your system allows it.
300102014	A Web Service request was received for a user, but the session for that user is not found.	Type: WARN:NIDP:WSF:014 **Cause:** The user's login has timed out and has been removed from the system. **Action:** The user must login again.
300102015	A Web Service request was received for a user, but the session for that user has insufficient data in it.	Type: WARN:NIDP:WSF:015 **Cause:** An internal error has occurred. **Action:** The user must login again.
300102016	A Web Service request was received for a user, but the Liberty User Profile object for that user is unavailable.	Type: WARN:NIDP:WSF:016 **Cause:** An internal error has occurred. **Action:** Make sure the administrator user has rights to read, write and create Liberty User Profile objects in the configuration data store.

Event Code	Message	Remedy
300102017	A Web Service request was received for a user, and attempt to read the requested attributes from the Liberty User Profile object was made, but an error occurred.	Type: WARN:NIDP:WSF:017 **Cause:** An internal error has occurred. **Action:** Evaluate the reason and take the appropriate actions.
300102018	A Web Service request was received for a user, While reading user data from an LDAP user object, a mismatch occurred because the LDAP attribute is multi-valued, but the Liberty attribute is single-valued.	Type: WARN:NIDP:WSF:018 **Cause:** A multi-valued LDAP attribute has been mapped to a single-valued Liberty attribute. **Action:** Change the attribute mapping.
300102019	The user used an X509 Certificate to authenticate and we tried to put the cert into the SecretStore as a Base64 DER encoded cert, but we got an encoding error from the security layer when trying to get the DER encoded cert. Result is that there will not be a X509 Certificate in Secret Store for this user.	Type: WARN:NIDP:WSF:019 **Cause:** The X509 certificate cannot be encoded. **Action:** Review the type of X509 certificates that are being used for authentication.
300102020	A SAMLAssertion was requested for a given user. While generating the SAMLAssertion an error occurred.	Type: WARN:NIDP:WSF:020 **Cause:** The SAMLAssertion cannot be created. **Action:** Review the reason for the failure and take appropriate actions.
300102021		Type: WARN:NIDP:WSF:021
300102022		Type: WARN:NIDP:WSF:022
300103001	The web service request did not return a response within the protocol timeout limit. Request abandoned.	Type: WARN:NIDP:WSC:001 **Cause:** The web service consumer waited for the web service request to return a response, but it did not during the allowed waiting period. **Action:** This waiting period may be increased by click Access Manager > Identity Servers > Edit > Liberty > Web Service Consumer, and setting the Protocol Timeout to a higher value.
300103002	An unexpected error happened in the web service consumer while processing a web service request.	Type: WARN:NIDP:WSC:002 **Cause:** There are various reasons why a web service request could fail. **Action:** Evaluate the reason and take appropriate actions.

Event Code	Message	Remedy
300103003	Web service consumer request pending data packet id is not available in request.	Type: WARN:NIDP:WSC:003 **Cause:** After user interaction, processing of the original request returns to the web service consumer. A data packet containing information about how to continue the request is cached on the web service consumer. The id of that packet must be passed through all redirections and requests associated with the user interaction. If that id is not available when the web service consumer regains control, then the request cannot continue. **Action:** Submit the request again.
300103004	The Web service consumer request pending data packet with the indicated id is not available in web service consumer's cache.	Type: WARN:NIDP:WSC:004 **Cause:** After user interaction, processing of the original request returns to the web service consumer. A data packet containing information about how to continue the request is cached on the web service consumer. The id of that packet must be passed through all redirections and requests associated with the user interaction. That id will be used to access the pending data packet when the web service consumer regains control. If the pending data packet with the corresponding id is no longer available on the system, then the request cannot continue. The data packet may have timed out. **Action:** Submit the request again.
300104049	Could not find NIDP PKIX Certificate Path Checker Class.	Type: WARN:NIDP:USERAUTH:049 **Cause:** PKIX Certificate Path Checker Class not found. **Action:** Warning message that PKIX Certificate Path Checker Class not found. This optional class is used to process custom certificate extensions. If required, this class needs to be in NIDP classpath. It may not be present on ESP.
300104050	Could not instantiate NIDP PKIX Certificate Path Checker Class.	Type: WARN:NIDP:USERAUTH:050 **Cause:** Incorrect class constructor. **Action:** Make sure the class has the right constructor.
300105001	No user Login Policy Check LDAP Extension method available on user store.	Type: WARN:NIDP:APP:001 **Cause:** User authenticated using X509. An additional check of the directory's user login policy needs to be made using an LDAP method extension. However, the directory indicated does not support the required LDAP extension method. **Action:** Make sure the LDAP extension method with OID 2.16.840.1.113719.1.39.42.100.25 is present in the user store. Versions 8.7.3 and greater of eDirectory should support this method.
300105002		Type: WARN:NIDP:APP:002
300105003		Type: WARN:NIDP:APP:003

Event Code	Message	Remedy
300105004		Type: WARN:NIDP:APP:004
300105005		Type: WARN:NIDP:APP:005
300105006		Type: WARN:NIDP:APP:006
300105007		Type: WARN:NIDP:APP:007
300105008		Type: WARN:NIDP:APP:008
300105009		Type: WARN:NIDP:APP:009
300105010		Type: WARN:NIDP:APP:010
300105011		Type: WARN:NIDP:APP:011
300105012		Type: WARN:NIDP:APP:012
300105013		Type: WARN:NIDP:APP:013
300105014		Type: WARN:NIDP:APP:014
300105015		Type: WARN:NIDP:APP:015
300105016		Type: WARN:NIDP:APP:016
300105017		Type: WARN:NIDP:APP:017
300105018		Type: WARN:NIDP:APP:018
300105019		Type: WARN:NIDP:APP:019
300105020		Type: WARN:NIDP:APP:020
300105021	Unable to delete unneeded Image Pool Image File.	Type: WARN:NIDP:APP:21 **Cause:** On startup, the NIDP Image Pool is synchronized from eDirectory to the file system. This allows HTML pages to access images from a well known file system structure. Part of synchronization process involves deleting from the file system images that no longer exist in eDirectory. Also, the reverse is true, images that are new to eDirectory and do not yet exist on the file system are created in directories that reflect the image set. File system errors may occur during this synchronization process if a file or directory cannot be deleted or created. **Action:** Ensure that no errant files are copied or directories manually created in the file system path [TOMCAT_HOME]/webapps/nidp/images/pool. Make sure the disk is not full.

Event Code	Message	Remedy
300105022	Unable to create a necessary directory for the Image Pool.	Type: WARN:NIDP:APP:22 **Cause:** On startup, the NIDP Image Pool is synchronized from eDirectory to the file system. This allows HTML pages to access images from a well known file system structure. Part of synchronization process involves deleting from the file system images that no longer exist in eDirectory. Also, the reverse is true, images that are new to eDirectory and do not yet exist on the file system are created in directories that reflect the image set. File system errors may occur during this synchronization process if a file or directory cannot be deleted or created. **Action:** Make sure the disk is not full.
300105023	Unable to create a necessary directory for the Image Pool.	Type: WARN:NIDP:APP:23 **Cause:** On startup, the NIDP Image Pool is synchronized from eDirectory to the file system. This allows HTML pages to access images from a well known file system structure. Part of synchronization process involves deleting from the file system images that no longer exist in eDirectory. Also, the reverse is true, images that are new to eDirectory and do not yet exist on the file system are created in directories that reflect the image set. File system errors may occur during this synchronization process if a file or directory cannot be deleted or created. **Action:** Make sure the disk is not full.
300105024	Unable to update the "last used" attribute of an identity object.	Type: WARN:NIDP:APP:24 **Cause:** Each time an identity object is accessed, the "last used" time is updated. This allows the system to track identities that have not been used for a configurable time period so that they may be deleted. **Action:** Make sure the administrator object for the Trust/Config data store has rights to the indicated directory context.
300105025	Unable to auto delete an identity object.	Type: WARN:NIDP:APP:25 **Cause:** Periodically, the IDP attempts to clean up (delete) identity objects that have not been used for a configurable period of time. If an old unused identity is found, an attempt will be made to delete it. If that delete fails, this error will be logged. **Action:** Make sure the administrator object for the Trust/Config data store has rights to the indicated directory context.

Event Code	Message	Remedy
300105027	No Filename specified in System property.	Type: WARN:NIDP:APP:27 **Cause:** Trying to read properties from file which is not specified in System property. **Action:** Make sure the properties file is passed in the appropriate system property.
300105028	Error trying to delete a CardSpace Issued Card Identity Object.	Type: WARN:NIDP:APP:28 **Cause:** When a CardSpace Managed Card that is backed by a Personal Card is issued, an Identity object is created to represent the "Federation" that allows that card to log into the IDP without supplying any additional credentials. For security reasons, the user may delete that Identity object, or that "federation," when the associated card becomes out of date or compromised. However, when the system attempted to delete the Identity object, the indicated error happened. **Action:** Examine the supplied error detail and take applicable actions.
300105029	Cannot load a custom LDAP Store Plugin module.	Type: WARN:NIDP:APP:29 **Cause:** The java.lang.Class.forName() method call failed to load the LDAP Store Plugin class. **Action:** Ensure a valid Java class file is available in Access Manager's class path for the referenced plugin class file.
300105030	Cannot instantiate a custom LDAP Store Plugin module.	Type: WARN:NIDP:APP:30 **Cause:** The java.lang.Class.newInstance() method call failed to instantiate the LDAP Store Plugin class. **Action:** Ensure a valid Java class file is available in Access Manager Appliance's class path for the referenced plugin class file. Also, ensure the LDAP Store Plugin has a zero parameter constructor.
300105031	A user store was configured with an unrecognized directory type.	Type: WARN:NIDP:APP:031 **Cause:** The configuration was manually modified to include an invalid directory type specifier. Or the configuration has been corrupted. Or there was no valid implementation of an LDAP Store Plugin for this directory type. **Action:** Examine the supplied error detail and take applicable actions.
300105036	Office365 assertion NameID value is null, check user <user name> attribute value.	**Cause:** User LDAP attribute value is empty in the user store. **Action:** Check the user attribute in configured user store. If NameID value is null, check user {1} attribute {0} value.
300106001		Type: WARN:NIDP:IDFF:001

Event Code	Message	Remedy
300106002		Type: WARN:NIDP:IDFF:002
300106003		Type: WARN:NIDP:IDFF:003
300106004		Type: WARN:NIDP:IDFF:004
300106005		Type: WARN:NIDP:IDFF:005
500102001	The authentication information for the user was successfully found.	Type: INFO:NIDP:WSF:001 Scenario: A Web Service request was made to query or modify user attributes. The user's authentication information was successfully found. See Also: 600102001
500102002	The Liberty User Profile object for the associated user was found in the configuration datastore.	Type: INFO:NIDP:WSF:002 Scenario: A Web Service request was made to query or modify user attributes. One of the data locations specified for the service is the Liberty User Profile object and that object was successfully found.
500102003	Created new user profile object.	Type: INFO:NIDP:WSF:003 Scenario: A request was made to query or modify user's attributes. A Liberty User Profile object did not yet exist for this user, so one was created.
500102004	Read data from user profile object.	Type: INFO:NIDP:WSF:004 Scenario: A Web Service request was made to query user attributes. One of the data locations specified for the service is the Liberty User Profile object and that object was successfully read. See Also: 600102002
500102005	Attempted to read data from the Liberty User Profile object, but it did not contain the requested data.	Type: INFO:NIDP:WSF:005 Scenario: A Web Service request was made to query user attributes. One of the data locations specified for the service is the Liberty User Profile object. That object was successfully accessed but did not contain the requested data.
500102006	Read data from attributes obtained when a remote authentication source pushed the attributes to the NIDP.	Type: INFO:NIDP:WSF:006 Scenario: When a user authenticates, the authentication entity can push user attributes to the NIDP as part of the response to the authentication. The NIDP remembers these attributes for the life of that user session. If one of the data locations specified for a Web Service is remote, then these attributes may be returned as part of a query. See Also: 600102005

Event Code	Message	Remedy
500102007	Read data by making a call to a remote service made available through a user authentication.	Type: INFO:NIDP:WSF:007 Scenario: A request was made to query a user's attributes. One of the data locations for the Web Service was remote. So, a request was made to a remote service to read attributes. See Also: 600102006
500102008	Completed building composite data that was read from all data locations for user.	Type: INFO:NIDP:WSF:008 Scenario: A request was made to query a user's attributes. If multiple data locations are specified for the Web Service, then attributes may be read from multiple data locations and then aggregated into a composite data structure. See Also: 600102007
500102009	Initiating a user interaction redirect.	Type: INFO:NIDP:WSF:009 Scenario: A request was made to query or modify user's attributes. Policy indicates that the user must be asked if the attribute operation is permitted. The request indicated that a redirect user interaction service should be used to perform user interaction, so redirection is being invoked using the redirection user interaction service protocol.
500102010	Initiating a user interaction call to a trusted user interaction service.	Type: INFO:NIDP:WSF:010 Scenario: A request was made to query or modify user's attributes. Policy indicates that the user must be asked if the attribute operation is permitted. The request indicated that a trusted user interaction service should be used to perform user interaction, so that service is being invoked using the trusted user interaction service protocol.
500102011	Read Credential Profile data from Novell Secret Store.	Type: INFO:NIDP:WSF:011 Scenario: A request was made to query data from a user's Credential Profile. The data was successfully read. See Also: 600102008
500102012	Read Credential Profile data from an extended user authentication object attribute.	Type: INFO:NIDP:WSF:012 Scenario: A request was made to query data from a user's Credential Profile. The data was read from an extended schema attribute on the user's authenticated user object. See Also: 600102010
500102013	Web service data write denied because the LDAP attribute plugin access for the named data item is read only!	Type: INFO:NIDP:WSF:013 Scenario: The system administrator has marked this data item as read only in the LDAP Attribute Plugin.

Event Code	Message	Remedy
500102014	Override not allowed. Cannot override existing data.	Type: INFO:NIDP:WSF:014 Scenario: The data that is being written already exists in the user's profile. Data override is not allowed so this data cannot be written.
500102015	Existing data changed since notChangedSince time.	Type: INFO:NIDP:WSF:015 Scenario: User profile data is marked with the last time the data changed. The query request indicated that it did not want the data written if the current data in the profile has been changed since an indicated time. The system determined that the current data in the profile has been changed since the time provided, so this data cannot be written.
500103001	Filled the user attribute request from data already in the web service consumer cache.	Type: INFO:NIDP:WSC:001 Scenario: When the WSC reads user attributes, it caches the results of each read. In this case, a subsequent request queried attributes already read, so they were provided from the WSC cache.
500103002	Web service consumer request complete.	Type: INFO:NIDP:WSC:002 Scenario: The WSC was asked to query or modify data for a given user. That request is complete.
500103003	Web service consumer request requires user interaction.	Type: INFO:NIDP:WSC:003 Scenario: The WSC was asked to query or modify data for a given user. The entity called to perform the operation indicated that the user must be asked if the attribute operation is acceptable.
500103004	User interaction policy and data values received.	Type: INFO:NIDP:WSC:004 Scenario: A Web Service request was made to query or modify user attributes. It was determined that the user must be asked if the attribute operation is acceptable. The user's answers have been returned to the NIDP.
500104002	Getting properties from file (informational)	Type: INFO:NIDP:USERAUTH:002 Scenario: Getting properties from file
500104007	X509 Authentication matched principal (informational)	Type: INFO:NIDP:USERAUTH:007 Scenario: X509 Authentication matched principal
500104013	No CRL/OCSP defined by the administrator	Type: INFO:NIDP:USERAUTH:013 Cause: No CRL/OCSP defined by the administrator
500104014	No CRL/OCSP found in the certificate.	Type: INFO:NIDP:USERAUTH:014 **Cause:** No CRL/OCSP found in the certificate **Action:** CRL/OCSP validations are enabled but no CRL/OCSP responder URL was defined by the administrator. CRL/OCSP URLs may be defined if needed.

Event Code	Message	Remedy
500104016	Could not fetch CRL from the local cache (informational)	Type: INFO:NIDP:USERAUTH:016 Scenario: Could not fetch CRL from the local cache, getting it from the CDP
500104048	Successfully loaded NIDP PKIX Certificate Path Checker Class (informational)	Type: INFO:NIDP:USERAUTH:048 Scenario: Successfully loaded NIDP PKIX Certificate Path Checker Class
500104113	Kerberos Principal match found in the user store (informational)	Type: INFO:NIDP:USERAUTH:113 Scenario: Kerberos Principal found in the user store
500105001	Forwarding HTTP request to cluster member.	Type: INFO:NIDP:APP:001 Scenario: A request was received on a cluster member that does not own the authentication information for the associated user. The request must be processed on the cluster member that does own the user authentication information, so the request is being forwarded to that cluster member.
500105002	Successfully initialized JNDI connections.	Type: INFO:NIDP:APP:002 Scenario: NIDP attempts to create JNDI connections to each user store replica during NIDP startup. In this case, NIDP was able to establish connections with the indicated host.
500105003	Failed X509 authentication due to Login Policy Check Extension Method evaluation.	Type: INFO:NIDP:APP:003 Scenario: The directory login policy for the indicated user denied login.
500105004	An recoverable error happened while forwarding a login request.	Type: INFO:NIDP:APP:004 Scenario: The request landed on the wrong cluster member. An attempt was made to proxy the request, but an error occurred! However, this ESP can process this request, so let execution proceed on this box.
500105005		Type: INFO:NIDP:APP:005
500105006		Type: INFO:NIDP:APP:006
500105007		Type: INFO:NIDP:APP:007
500105008		Type: INFO:NIDP:APP:008
500105009		Type: INFO:NIDP:APP:009
500105010		Type: INFO:NIDP:APP:010
500105011		Type: INFO:NIDP:APP:011
500105012		Type: INFO:NIDP:APP:012
500105013		Type: INFO:NIDP:APP:013
500105014		Type: INFO:NIDP:APP:014
500105015		Type: INFO:NIDP:APP:015

Event Code	Message	Remedy
500105016		Type: INFO:NIDP:APP:016
500105017		Type: INFO:NIDP:APP:017
500105018		Type: INFO:NIDP:APP:018
500105019		Type: INFO:NIDP:APP:019
500105020		Type: INFO:NIDP:APP:020
500105021		Type: INFO:NIDP:APP:021
500105022		Type: INFO:NIDP:APP:022
500105023		Type: INFO:NIDP:APP:023
500105024		Type: INFO:NIDP:APP:024
500105025		Type: INFO:NIDP:APP:025
500105026		Type: INFO:NIDP:APP:026
500105027		Type: INFO:NIDP:APP:027
500105028		Type: INFO:NIDP:APP:028
500105029		Type: INFO:NIDP:APP:029
500105030		Type: INFO:NIDP:APP:030
500105031		Type: INFO:NIDP:APP:031
500105032		Type: INFO:NIDP:APP:032
500105033		Type: INFO:NIDP:APP:033
500105034		Type: INFO:NIDP:APP:034
500105035		Type: INFO:NIDP:APP:035
500105036		Type: INFO:NIDP:APP:036
500105037		Type: INFO:NIDP:APP:037
500105038		Type: INFO:NIDP:APP:038
500105039		Type: INFO:NIDP:APP:039
500105040		Type: INFO:NIDP:APP:040
500105041		Type: INFO:NIDP:APP:041
500105042		Type: INFO:NIDP:APP:042
500105043		Type: INFO:NIDP:APP:043
500105044		Type: INFO:NIDP:APP:044
500105045		Type: INFO:NIDP:APP:045

Event Code	Message	Remedy
500105046	The specified identity object was deleted because it was not used for a configurable time period.	Type: INFO:NIDP:APP:046 Scenario: Periodically, the IDP attempts to clean up (delete) identity objects that have not been used for a configurable period of time. If an old unused identity is found, an attempt will be made to delete it. When this delete succeeds, this message will be logged.
500106001		Type: INFO:NIDP:IDFF:001
500106002		Type: INFO:NIDP:IDFF:002
500106003		Type: INFO:NIDP:IDFF:003
500106004		Type: INFO:NIDP:IDFF:004
500106005		Type: INFO:NIDP:IDFF:005
500106006		Type: INFO:NIDP:IDFF:006
500106007		Type: INFO:NIDP:IDFF:007
500106008		Type: INFO:NIDP:IDFF:008
600102001	Verbose user authentication information.	Type: DEBUG:NIDP:WSF:001 Scenario: Adds verbose authentication data to the fact that the user associated with the attribute request was found in the internal databases of the web service provider. See Also: 500102001
600102002	Verbose user authentication information, attribute select string, and data.	Type: DEBUG:NIDP:WSF:002 Scenario: A Web Service request was made to query user attributes. One of the data locations specified for the service is the Liberty User Profile object. The data listed in this message was successfully read for the indicated user using the indicated XPath. See Also: 500102004
600102003	Read single-valued attribute from user authentication LDAP object.	Type: DEBUG:NIDP:WSF:003 Scenario: A Web Service request to query user attribute data was received. One of the data locations was LDAP. This message displays the value read from the indicated LDAP attribute for the indicated user.
600102004	Read multi-valued attribute from user authentication LDAP object.	Type: DEBUG:NIDP:WSF:004 Scenario: A Web Service request to query user attribute data was received. One of the data locations was LDAP. This message displays the value read from the indicated LDAP attribute for the indicated user.

Event Code	Message	Remedy
600102005	Verbose user authentication and attribute information.	Type: DEBUG:NIDP:WSF:005 Scenario: When a user authenticates, the authenticating entity can push user attributes to the NIDP as part of the response to the authentication. The NIDP remembers these attributes for the life of that user session. If one of the data locations specified for a Web Service is remote, then these attributes may be returned as part of a query. See Also: 500102006
600102006	Adds verbose user and attribute information to attributes read from a remote service whose description was obtained at authentication time.	Type: DEBUG:NIDP:WSF:006 Scenario: A request was made to query a user's attributes. One of the data locations for the Web Service was remote. So, a request was made to a remote service to read attributes. See Also: 500102007
600102007	Adds verbose user and attribute information to the final aggregated result of a web service query!	Type: DEBUG:NIDP:WSF:007 Scenario: A request was made to query a user's attributes. If multiple data locations are specified for the Web Service, then attributes may be read from multiple data locations and then aggregated into a composite data structure. See Also: 500102008
600102008	Adds verbose data to reading Credential Profile data from Novell Secret Store.	Type: DEBUG:NIDP:WSF:008 Scenario: A request was made to query data from a user's Credential Profile. The data was successfully read. See Also: 500102011
600102009	The user successfully logged into Novell Secret Store using SAML/SASL.	Type: DEBUG:NIDP:WSF:009 Scenario: To access secrets from Novell Secret Store, the user must authenticate to Novell Secret Store.
600102010	Adds verbose data to reading Credential Profile data from an extended user authentication object attribute.	Type: DEBUG:NIDP:WSF:010 Scenario: A request was made to query data from a user's Credential Profile. The data was read from an extended schema attribute on the user's authenticated user object. See Also: 500102012
600105001	Do not need to proxy HTTP request to other cluster member. Well known URL that does not require the use of a proxy.	Type: DEBUG:NIDP:APP:001 Scenario: The request is one of a well known list of request types that may be processed on any cluster member, so it does not need to be forwarded to another cluster member.

Event Code	Message	Remedy
600105002	Do not need to proxy HTTP request to other cluster member. This cluster member can handle requests for this user.	Type: DEBUG:NIDP:APP:002 Scenario: The request arrived at the cluster member that owns the authentication information for the user. The request may have come straight from the router to this cluster member, or the request may have been forwarded here by another cluster member.
600105003	Obtained IP address of cluster member handling this users requests from URL parameter.	Type: DEBUG:NIDP:APP:003 Scenario: Each request must be processed on the cluster member that owns the user authentication information. The IP address of that cluster member was found in a URL parameter.
600105004	Obtained IP address of cluster member handling this users requests from HTTP cookie.	Type: DEBUG:NIDP:APP:004 Scenario: Each request must be processed on the cluster member that owns the user authentication information. The IP address of that cluster member was found in an HTTP cookie.
600105005	Obtained IP address of cluster member handling this user's requests by asking cluster members which one handles this user session.	Type: DEBUG:NIDP:APP:005 Scenario: Each request must be processed on the cluster member that owns the user authentication information. The IP address of that cluster member was found by asking all cluster members which one knew about the user's session.
600105006	Must proxy HTTP request to other cluster member.	Type: DEBUG:NIDP:APP:006 Scenario: Each request must be processed on the cluster member that owns the user authentication information. It has been determined that this cluster member is not the correct cluster member to process this request, so the request must be forwarded to another cluster member.
600105007	Response of proxy HTTP request.	Type: DEBUG:NIDP:APP:007 Scenario: Each request must be processed on the cluster member that owns the user authentication information. It was determined that this cluster member is not the correct cluster member to process this request, so the request was forwarded to another cluster member. The results of the request, as processed on the other cluster member, are displayed here.
600105008	Successfully obtained SOAP response document.	Type: DEBUG:NIDP:APP:008 Scenario: A SOAP request was made and a response was expected, the response was successfully obtained.
600105009		Type:DEBUG:NIDP:APP:009
600105010		Type: DEBUG:NIDP:APP:010
600105011		Type: DEBUG:NIDP:APP:011

Event Code	Message	Remedy
200104401	Login failed. Please try again.	Rule group is not associated to Risk-Based authentication class. Map a rule group to the class.
200104403	Authentication failed.	Authentication failed. The geolocation rule is enabled but the geolocation provider is not configured or is configured incorrectly.
500104400	Access denied. Contact your administrator	User is denied login because the risk score is high.
500104402	Access denied. Contact your administrator	No risk level is defined for this risk score.
200104067	The target domain is unknown. Contact your administrator.	**Cause**: URL is not configured as a whitelist domain or it is invalid. **Action**: If the URL is valid and required, contact the administrator to get it configured in the redirection whitelist.

Linux Access Gateway Appliance(045)

Component 045

Event Code	*Description*	Remedy
[1-9]04501000	Multi-homing	See the string value in the message for a description of the cause.
[1-9]04502000	Service manager	See the string value in the message for a description of the cause.
[1-9]04503000	Browser request processing	See the string value in the message for a description of the cause.
[1-9]04504000	Authentication processing	See the string value in the message for a description of the cause.
[1-9]04505000	Authorization processing	See the string value in the message for a description of the cause.
[1-9]04506000	Identity Injection processing	See the string value in the message for a description of the cause.
[1-9]04507000	Form Fill processing	See the string value in the message for a description of the cause.
[1-9]04508000	Caching	See the string value in the message for a description of the cause.
[1-9]04509000	Processing of Web server responses and of responses to browser requests	See the string value in the message for a description of the cause.
[1-9]04511000	Rewriter processing	See the string value in the message for a description of the cause.
[1-9]04512000	SOAP back channel processing	See the string value in the message for a description of the cause.

Event Code	*Description*	Remedy
[1-9]04513000	Device communication channel (VCC)	See the string value in the message for a description of the cause.
[1-9]04514000	VM controller processing	See the string value in the message for a description of the cause.
[1-9]04515000	Connection management	See the string value in the message for a description of the cause.
[1-9]04516000	Core utilities (VXE)	See the string value in the message for a description of the cause.
[1-9]04517000	Data Stream processing	See the string value in the message for a description of the cause.
[1-9]04518000	SSL processing	See the string value in the message for a description of the cause.
[1-9]04519000	Command processing	See the string value in the message for a description of the cause.
[1-9]04520000	Profiler	See the string value in the message for a description of the cause.
[1-9]04521000	Proxy start	See the string value in the message for a description of the cause.
[1-9]04522000	Audit event processing	See the string value in the message for a description of the cause.

Access Gateway Service (046)

Component 046

- Subgroup 00: URL Request Processing
- Subgroup 01: Authorization Processing
- Subgroup 02: Identity Injection Processing
- Subgroup 03: Form Fill Processing
- Subgroup 30: Web Server Communication Processing
- Subgroup 50: Administration Request Processing
- Subgroup 51: Statistics
- Subgroup 52: Health
- Subgroup 53: Alerts Processing
- Subgroup 54: Configuration Processing
- Subgroup 55: Initialization-Termination Processing

Event Code	Description	Remedy
URL Request Processing (00)		

Event Code	Description	Remedy
304600404	Authentication Request: Unknown Contract	**Cause:** An unknown contract was received from the Embedded Service Provider. This can happen if the configuration of the Identity Server and Access Gateway are not synchronized. **Action:** Check to see if the Access Gateway or the Identity Server need to be updated. If their status is current, make a small change to both and update their configuration.
504600000	URL Accessed	A request for access to an unprotected URL has been received.
504600100	Protected Resource Accessed	A request for access to a protected URL has been received.
504600400	Authentication Request: Successful	The user authenticated successfully.
504600401	Login Request: Redirect To ESP	The authentication request was redirected to the Embedded Service Provider
504600402	Authentication Request: Set Cookie	The request has been redirected to set the cookie.
504600403	Authentication Request: Redirect URL with Cookie	The original URL request has been redirected to the Embedded Service Provider with a cookie.
504600405	Authentication Request: NRL Request	The protected resource is configured for non-redirected login.
604600001	URL Accessed: Trace Summary	This event accesses the URL trace summary.
604600002	URL Accessed: Scheme Redirect	The URL accessed on wrong scheme is redirected.
604600003	URL Accessed: Pinned	The URL in the PIN list is accessed.
604600301	Session Broker: Cookie Not Found	The session broker returns the status of cokkie not found.
604600302	Session Broker: Add User	The session broker requests to add user.
604600303	Session Broker: Get Cookie	The session broker requests cookie.
604600304	Session Broker: Delete User	The session broker deletes user sent from SOAP request.
604600306	Session Broker: Update User	The session broker updates the user sent from SOAP request.
604600307	Session Broker: Cookie Found	The session broker returned requested cookie.
604600308	Session Broker: Add User SOAP Request	The session broker adds the user sent from SOAP request.
604600309	Session Broker: User Added	The session broker adds user request which are successfully processed.
604600310	Session Broker: Delete User Successful	The session broker deletes the user successfully.

Event Code	Description	Remedy
604600311	Session Broker: Delete User Failed	The session broker failed to delete user.
Authorization Processing (01)		
204601102	Policy Configuration Reply: Policy Error	**Cause:** An error was detected while processing a policy configuration request. **Action:** Check the health of the configuration database. If it is unhealthy, repair it or restore it from a backup.
204601302	Policy Evaluation Reply: Policy Error	**Cause:** An error was detected while processing a policy evaluation request. **Action:** Verify that the Embedded Service Provider and the proxy service are running.
504601003	ACL Policy Configuration Request	ACL configuration request is being processed.
504601100	Policy Configuration Reply: Success	The Authorization policy has been configured successfully.
504601203	Policy Evaluation Request	A policy evaluation request has been received; the evaluation has started.
504601300	Policy Configuration Reply: Access allowed, no match	The Authorization policy evaluation results allowed access due to policy default action.
504601301	Policy Configuration Reply: Access allowed	The Authorization policy evaluation results allowed access.
504601302	Policy Configuration Reply: Access denied	The Authorization policy evaluation results denied access.
Identity Injection Processing (02)		
204602102	Policy Configuration Reply: Policy Error	**Cause:** An error was detected while processing a policy configuration request. **Action:** Check the health of the configuration database. If it is unhealthy, repair it or restore it from a backup.
204602302	Policy Evaluation Reply: Policy Error	**Cause:** An error was detected while processing a policy evaluation request. **Action:** Verify that the Embedded Service Provider and the proxy service are running.
504602100	Policy Configuration Reply: Success	The Identity Injection policy has been configured successfully.
504602300	Policy Evaluation Reply: Inject Authentication Header	This policy injects an authentication header
504602301	Policy Evaluation Reply: Inject Custom Headers	This policy injects custom headers.
504602302	Policy Evaluation Reply: Inject Query Parameters	This policy injects query parameters.
Form Fill Processing (03)		

Event Code	Description	Remedy
204603101	Policy Configuration Reply: No Policy ID	The policy ID is not included with policy configuration request.
204603102	Policy Configuration Reply: Policy Error	**Cause:** An error was detected while processing a policy configuration request. **Action:** Check the health of the configuration database. If it is unhealthy, repair it or restore it from a backup.
204603302	Policy Evaluation Reply: Policy Error	**Cause:** An error was detected while processing a policy evaluation request. **Action:** Verify that the Embedded Service Provider and the proxy service are running.
204603304	Policy Evaluation Reply: Parse Error: Unknown field	**Cause:** A parsing error was detected while processing a policy evaluation request. **Action:** Check the Form Fill policy and make sure it matches the form.
504603100	Policy Configuration Reply: Success	The Form Fill policy has been configured successfully.
504603300	Policy Evaluation Reply: Success	The Form Fill policy evaluation was successful.
504603301	Policy Evaluation Reply: No Policy	The Form Fill policy was not found.
504603400	Get User Attributes	A request has been sent to get user attributes.
504603401	Set User Attributes	A request has been sent to set user attributes.
Administration Request Processing (50)		
204650002	DCC Message Processing	These events will processes the DCC messages.
504650002		
604650002		
704650002		
204650003	JCC	The information is related to sending and processing JCC requests.
204650005	Device Information Requests	These events will process the request of the device information.
504650005		
604650005		
704650005		
204650001	Command Processing	The Administration Console initiates the log events pertaining to processing commands.
504650001		
604650001		
704650001		
304650004	Service Information Requests	The service information requests are processed.
604650004		

Event Code	Description	Remedy
504650010	Start	The log events pertaining to a Start command received from the Administration Console.
504650011	Stop	The log events pertaining to a Stop command received from the Administration Console.
504650012	Restart	The log events pertaining to a Restart command received from the Administration Console.
504650013	Refresh Policy	The log events pertaining to a Refresh Policy command received from the Administration Console.
504650014	Cache Clear	The log events pertaining to a Cache Clear command received from the Administration Console.
504650015	IP Scan	The log events pertaining to an IP Scan command received from the Administration Console.
Statistics (51)		
204651001	Statistics Request Processing	The log events pertaining to the processing of a Statistics request from device manager.
304651001		
504651001		
604651001		
704651001		
504651000	Statistics	The log of current statistics requested by device manager.
Health (52)		
204652001	Health Request Processing	The log events pertaining to the processing of a health request from device manager.
304652001		
604652001		
704652001		
504652000	Health	The log of current health as requested by device manager.

Policy Engine (008)

Component 008

- Subgroup 01: Engine
- Subgroup 02: Condition Handler
- Subgroup 03: Action Handler
- Subgroup 04: Configure Information Context

- Subgroup 05: Information Context
- Subgroup 06: Response Context

* = any Sub group

Event Code	*Description*	**Remedy**
100801001 100802001 100803001 100804001 100805001 100806001	Error No Memory: Memory allocation failed.	**Cause:** Low system memory. Resource allocation failed. **Action:** Determine cause for low system memory and resolve.
200801002 200802002 200803002 200804002 200805002 200806002	Error Bad Data: Policy configuration contains an invalid policy parameter list enumerative value.	**Cause:** The Administration Console has produced an invalid policy configuration document. **Cause:** Policy configuration document has been corrupted. **Action:** Take any or all of the following actions: 1. Submit the log file to Novell Support to aid in determining and fixing the source of the problem. 2. Back up to a previously working policy configuration until the problem has been fixed. 3. Examine the policy configuration document (available in PEP trace entries) to determine the erroneous policy document element and remove the corresponding policy statement from your policy configuration until a fix for the problem is available.

Event Code	*Description*	**Remedy**
200801003 200802003 200803003 200804003 200805003 200806003	Error Configuration. The policy configuration is syntactically incorrect or malformed.	**Cause:** The Administration Console has produced an invalid policy configuration document. **Cause:** Policy configuration document has been corrupted. **Action:** Take any or all of the following actions: 1. Submit the log file to Novell Support to aid in determining and fixing the source of the problem. 2. Back up to a previously working policy configuration until the problem has been fixed. 3. Examine the policy configuration document (available in PEP trace entries) to determine the erroneous policy document element and remove the corresponding policy statement from your policy configuration until a fix for the problem is available.
200801004 200802004 200803004 200804004 200805004 200806004	General Failure: Internal software error.	**Cause:** Unexpected exception caught during policy evaluation. **Action:** Submit log file to Novell Support for analysis and problem resolution.

Event Code	*Description*	**Remedy**
200801072 200802072 200803072 200804072 200805072 200806072	Interface Unavailable: Invalid InformationContext or ResponseContext enumerative value.	**Cause:** The Administration Console has produced an invalid policy configuration document. Invalid PolicyTypeSpec schema. **Cause:** Policy configuration document has been corrupted. **Action:** Take any or all of the following actions: 1. Submit the log file to Novell Support to aid in determining and fixing the source of the problem. 2. Back up to a previously working policy configuration until the problem has been fixed. 3. Examine the policy configuration document (available in PEP trace entries) to determine the erroneous policy document element and remove the corresponding policy statement from your policy configuration until a fix for the problem is available.
200801073 200802073 200803073 200804073 200805073 200806073	Data Unavailable: Policy Engine could not obtain needed information to complete policy evaluation.	**Cause:** Inaccessible user store or database. **Action:** Ensure user store or database is available. **Cause:** Network connectivity problems. **Action:** Ensure network is operational.
200801074 200802074 200803074 200804074 200805074 200806074	Illegal State: Policy Engine caught NullPtrException during policy configuration or evaluation.	**Cause:** Unexpected software exceptions. **Action:** Submit log to Novell Support for analysis and resolution.
200801075 200802075 200803075 200804075 200805075 200806075	Illegal Argument: Internal software error.	**Cause:** Invalid method argument received. **Action:** Submit log to Novell Support for analysis and resolution.

Event Code	*Description*	Remedy
300801071 300802071 300803071 300804071 300805071 300806071	Evaluation Canceled: Active policy evaluation canceled.	**Cause:** May occur during system shutdown. **Action:** If not caused by system shutdown, submit log to Novell Support for analysis and resolution.
500801000 500802000 500803000 500804000 500805000 500806000	Success: Policy operation completed without error.	**Cause:** Policy Evaluation. **Action:** No Action. Informational only.
500801005 500802005 500803005 500804005 500805005 500806005	Operation Pending: Policy operation is in progress	**Cause:** Policy Evaluation. **Action:** No Action. Informational only.
500803064	Permit Action: Policy evaluation rendered a Permit Action.	**Cause:** Permit action executed. **Action:** No Action. Informational only.
500803065	Deny Action: Policy evaluation rendered a Deny Action.	**Cause:** Deny action executed. **Action:** No Action. Informational only.
500803066	Obligation Action: Policy evaluation rendered an Obligation Action.	**Cause:** Obligation action executed. **Action:** No Action. Informational only.
500801067 500802067 500803067 500804067 500805067 500806067	No Action: Policy evaluation rendered no Action.	**Cause:** No action was executed during a policy evaluation. **Action:** No Action. Informational only.
500802068	Condition False: Policy condition returned FALSE.	**Cause:** Policy Evaluation. **Action:** No Action. Informational only.

Event Code	*Description*	**Remedy**
500802069	Condition True: Policy condition returned TRUE.	**Cause:** Policy Evaluation. **Action:** No Action. Informational only.
200802070	Condition Unknown. Policy configuration contains an unsupported condition handler definition.	**Cause:** The Administration Console has produced an invalid policy configuration document. **Cause:** Policy configuration document has been corrupted. **Action:** Take any or all of the following actions: 1. Submit the log file to Novell Support to aid in determining and fixing the source of the problem. 2. Back up to a previously working policy configuration until the problem has been fixed. 3. Examine the policy configuration document (available in PEP trace entries) to determine the erroneous policy document element and remove the corresponding policy statement from your policy configuration until a fix for the problem is available.

SOAP Policy Enforcement Point (011)

The SOAP Policy Enforcement Point (PEP) interface is used by the NetWare and Access Gateways for policy evaluation.

Component 011

- Subgroup 01: General/Configuration
- Subgroup 02: Authorization PEP
- Subgroup 03: Identity Injection PEP
- Subgroup 04: Form Fill PEP

Messages are logged to the catalina.out for trace and application level logging when Identity Server logging is enabled.

Event Code	Description	Remedy
General/Configuration		
501101010	Start Policy Soap Handler	Policy Soap Message Handler received start command. **Cause:** Embedded Service Provider has been started **Action:** None. Informational message only.

Event Code	Description	Remedy
501101011	Stop Policy Soap Handler	Policy Soap Message Handler received stop command. **Cause:** Embedded Service Provider has been stopped **Action:** None. Informational message only.
101101012	Policy Evaluator Not Running	The Policy Evaluator has been stopped. **Cause:** The Embedded Service Provider has been stopped by an administrator **Action:** Restart the Embedded Service Provider for the device.
101101013	General Failure	General failure processing policy request. **Cause:** Most often caused by incorrectly formatted XML. **Action:** Check catalina.out for stack trace and possibly more detailed information regarding the failure.
501101020	Request Received	Soap request received. **Cause:** Informational message which logs the type of request received **Action:** None. Informational message used for checking soap handler interactions.
501101021	Response Sent	Soap response sent. **Cause:** Informational message regarding soap response to a request **Action:** None. Informational message used for checking soap handler interactions.
101101022	Unsupported request received	A NXPES command other than configure, evaluate or terminate was received. **Cause:** The policy engine revision is incompatible with the application. **Action:** Validate the software installation.
201101023	Unrecognized Policy Identifier	Policy evaluation was requested for an unknown policy. **Cause:** The policy identifier known to the Access Gateway is stale. **Action:** Most often, this problem is detected by the Access Gateway and the policies are reconfigured. If the problem persists, send an Apply or Apply Changes to the device from the CLI or Administrative Console.

Event Code	Description	Remedy
501101030	Configure Success	Successful policy configuration. **Cause:** Policy configuration succeeded **Action:** None. Informational message used for checking policy configuration.
201101030	Configure Warning	Policy Configuration Warning. **Cause:** Policy configuration request reported a problem in retrieving configuration data from the config store **Action:** Check the policy definitions in the Administration Console to ensure the configuration store is working properly, then reapply the configuration to the device.
101101031	Configure Failure	The policy requested is malformed or causes an exception during the configuration process. **Cause:** This is accompanied with a possible reason for the failure. **Action:** Check the policy configuration in the administrative console and reapply the configuration to the device.
501101032	Configure - Empty Policy Set	The set of policies requested either do not apply to the policy enforcement point or the set of policies do not match the categories selected in the policy enforcement list. **Cause:** This may be normal operation. **Action:** If a policy is expected, check the category of the policy and make sure the policy is enabled for the device.
501101040	Terminating policy	The set of policies represented by the policy ID are no longer needed and will be removed from the operating policy set. **Cause:** This happens each time a configuration is applied to the device. **Action:** None. This is an informational message only.
501101050	Evaluating policy	An evaluation request has been received for the set of policies represented by the policy ID. **Cause:** This happens at least once per user session per configured policy enforcement point. **Action:** None. This is an informational message only.

Event Code	Description	Remedy
501101051	Policy Evaluation - Invalid User Error	User session received for policy evaluation was not found or contains invalid data. **Cause:** The Identity Service Provider which authenticated the user is not accessible from the Embedded Service Provider. **Action:** Most often, this error will automatically restart the user identification process for the Access Gateway. If that does not occur: **Administrator:** If problem persists, check health status of Identity Service Provider and take appropriate action. **End User:** Retry request. If not redirected to the Identity Service Provider, force a refresh of the current browser page and the Access Gateway/Embedded Service Provider will reinitiate the authentication process.
501101052	Policy Evaluation - Information Query Error	The Policy Evaluator is unable to gain access to information required by the policy. **Cause:** This is accompanied with a possible reason for failure. **Action:** As the administrator, check the health status of Identity Service Provider and take appropriate action.
501101053	Policy Evaluation - WSC Query Error	An attempt to use the WSC query mechanism of the ESP failed, the requested policy data is unavailable. **Cause:** This is accompanied with a possible reason for failure. **Action:** As the administrator, check the health status of Identity Service Provider and take appropriate action.

Event Code	Description	Remedy
501101054	Policy Evaluation - Cluster Data Query Error	Attempt to retrieve user session data from ESP cluster member failed. **Cause:** The Embedded Service Provider which authenticated the user may not be accessible from the Embedded Service Provider evaluating the policy. **Action:** Most often, this error will automatically restart the user identification process for the Access Gateway. If that does not occur: **End User:** Close browser and retry request. **Administrator:** Check the health status of Embedded Service Provider referenced by IP address in the log and take appropriate action.
501101055	Policy Evaluation - Cluster Query Retry Count	Informational message containing the number of retries the ESP has made to request policy information from another cluster member. **Cause:** The Embedded Service Provider which authenticated the user may not be accessible from the Embedded Service Provider evaluating the policy. **Action:** None, this is an informational message only.
Authorization PEP		
501102050	Policy Evaluation Trace	Trace of an individual policy evaluation. **Cause:** Policy evaluation. **Action:** None. Informational message used for checking policy evaluation.
Identity Injection PEP		
501103050	Policy Evaluation Trace	Trace of an individual policy evaluation. **Cause:** Policy evaluation. **Action:** None. Informational message used for checking policy evaluation.
Form Fill PEP		
501104050	Policy Evaluation Trace	Trace of an individual policy evaluation. **Cause:** Policy evaluation. **Action:** None. Informational message used for checking policy evaluation.

Backup and Restore (010)

Backup and restore are invoked by script files:

- defbkparm.sh: Created by install. This has the default values for the scripts.
- getparams.sh: Prompts administrator for information needed to do the backup or restore operation.
- ambkup.sh: Script to run to perform a backup.
- amrestore.sh: Script to run to perform a restore.

Other programs used by backup and restore:

- ICE: This is the Novell eDirectory utility to import and export LDIF file in and out of eDirectory.
- ldifReverse: This is a program that reverses the order of the records in the LDIF file exported from eDirectory. Reversing the order of records allows the LDIF file to be imported without errors.
- certtool.jar: This is a eDirectory certificate utility that backs up and restores the CA key, server keys, and trusted roots to a zip file.

Component 010

- Subgroup 01: Backup
- Subgroup 02: Restore
- Subgroup 03: certtool (certificate backup and restore)

Messages are logged to the ambkup.log file.

Event Code	*Description*	**Remedy**
	Backup	
201001001	Backup failed to export data from the configuration store.	**Cause:** The ICE utility failed to export directory information to an LDIF file. **Action:** Make sure that ICE is in the proper location (Linux: `/opt/novell/eDirectory/bin`). **Action:** Make sure that the host IP address, port, administrator, password are all correct. **Action:** Make sure the back up file is writable
201001002	Backup failed to format data for a successful restore.	**Cause:** The ldifReverse utility failed to sort the LDIF records. **Action:** Make sure that ldifReverse is in the proper location (Same directory as backup command). **Action:** Make sure the back up file is writable **Action:** Check for the backup file you specified with "_pre" appended to the file name. If the file exists, run the following command: `ldifReverse bkupfile_pre bkupfile` Replace `bkupfile` with the filename you specified for the backup file. It should create `bkupfile` which is the desired back up file.

Event Code	*Description*	Remedy
201001003	Backup failed to export certificates to the backup zip file.	**Cause:** The certtool utility failed to export the certificates to a zip file. **Action:** Make sure that `certtool.jar` is in the proper location (Same directory as backup command). **Action:** Make sure the back up file is writable. **Action:** Manually export the certificates to a zip file: `java -Djava.library.path=/opt/novell/lib -jar certtool.jar -edirTree your_tree -edirIP 000.000.000.000 -edirServer cn=!ServerName.0=novell -edirUser cn=admin.o=novell -edirPwd secret -bkup -file ServerName _20060828_0930.zip -pwd certsecret -trcontainer trustedRoots.access ManagerContainer.novell -caName "your_tree CA"`
	Restore	
201002001	Backup file does not exist.	**Cause:** The backup file does not exist. The name of the backup file specified in answer to the prompt should not include the final the `.ldif` or `.zip` extension. **Action:** Specify the correct name of the back up file.
201002002	Backup file does not appear to be valid.	**Cause:** An simple analysis of the backup file indicates that the LDIF file specified backup file (with `.ldif` appended to the name) is not a valid backup file. **Action:** Make sure to specify a backup file that was created by the Access Manager Appliance Backup utility.
201002003	Restore failed to access the configuration store.	**Cause:** The ICE utility failed to access the eDirectory configuration store. **Action:** Make sure that ICE is in the proper location (Linux: `/opt/novell/eDirectory/bin`). Make sure that the host IP address, port, administrator, password are all correct.
201002004	Restore failed to format the current configuration store data.	**Cause:** Restore was not able to save a current copy of the configuration store. A current copy of the config store is saved before the import in case the import fails. **Action:** Make sure that ldifReverse is in the proper location (Same directory as backup command).
201002005	Restore failed to prepare the configuration store for data import.	**Cause:** ICE failed. Unknown reason because it has previously been invoked successfully in the restore script.
201002006	Restore failed to prepare the configuration store for data import.	**Cause:** ICE failed. Unknown reason because it has previously been invoked successfully in the restore script.

Event Code	*Description*	**Remedy**

101002007	Restore failed to restore the backup data.	**Cause:** ICE failed. Unknown reason because it has previously been invoked successfully in the restore script.
		
		Action: Check the configuration store for the following container:
		
		`ou=accessManagerContainer,o=novell`
		
		If it is not there, locate the `recover.ldif` file. It should be in the directory where you ran the restore command. Run ICE to recover the configuration store to the state it was in before you attempted the restore. Enter the following command:
		
		`/opt/novell/eDirectory/bin/ice -SLDIF -f recover.ldif -C -n -DLDAP -sxxx.xxx.xxx.xxx -p636 -k -dcn=admin, o=novell -wadmin_password -F`

101002008	Failed to restore certificate from backup file.	**Cause:** The java program restores the certificate failed. The java program is certtool.jar which provides command line access to various eDirectory certificate functions. **Action:** See the log file (`ambkup.log`) for more specific details. The log file contains a listing of relevant parameters with each error message. Assuming the back up from which you are trying to restore was successful, failure to restore is probably an incorrectly supplied parameter. Enter the following command: `JAVA -classpath vcdnbkup.jar:cert tool.jar com.novell.nids.bkuputil. Util -userid cn=admin,o=novell -pwd secret -vcdnUser`
101002009	Failed to reconfigure VCDN user objects.	**Cause:** The VCDN user objects were not restored with their passwords. Device Manager will not start up until the passwords have been properly set. **Action:** This is accompanied with an error x01004xxx. Please refer to that error.
	certtool utility	
201003002	IP address is missing.	**Cause:** The certtool.jar was launched without the -edirIP option. A script file might have been incorrectly modified. **Action:** Make sure the -edirIP option is specified in the script when it launches the certtool utility.
201003005	eDirectory user id missing.	**Cause:** The certtool.jar was launched without the -eDirUser option. A script file might have been incorrectly modified. **Action:** Make sure the -edirUser cn=admin.o=novell option is specified in the script when it launches the certtool utility.

Event Code	*Description*	Remedy
201003006	eDirectory user password missing.	**Cause:** The certtool.jar was launched without the -edirPwd option. A script file may have been incorrectly modified. **Action:** Make sure the -edirPwd option is specified in the script when it launches the certtool utility.
201003009	File name missing.	**Cause:** The certtool.jar was launched without the -file (name of backup file) option. A script file may have been incorrectly modified. **Action:** Make sure the -file option is specified in the script when it launches the certtool utility.
201003011	Encryption password missing.	**Cause:** The certtool.jar was launched without the -pwd option. A script file may have been incorrectly modified. **Action:** Make sure the -pwd option is specified in the script when it launches the certtool utility.
201003013	Name of trusted root container missing.	**Cause:** The certtool.jar was launched without the -trContainer (trusted root container) option. A script file may have been incorrectly modified. **Action:** Make sure the -trcontainer option is specified in the script when it launches the certtool utility.
201003040	Failed to open backup file for writing.	**Cause:** Backup was unable to create or access the backup file in which to save certificate information. **Action:** Ensure that user running backup sufficient rights.
201003041	Failed to retrieve certificate names from eDirectory.	**Cause:** A PKI or eDirectory error. **Action:** This error will be accompanied by an error string.
201003042	Failed to retrieve certificate xxxx from eDirectory.	**Cause:** The certtool failed to retrieve the certificate identified in the error. Problems have been seen trying to export certificate with pending CSRs. **Action:** This error will be accompanied by an error string.
201003043	Failed to write certificate xxxx to backup file.	**Cause:** The certificate identified in the error message did not get saved to the backup file. **Action:** An exception string included in the message my provide additional information.
301003044	Error closing backup.	**Cause:** Likely will not cause a problem. **Action:** Try extracting the contents of the zip file created by backup to verify the integrity of the zip file.
201003045	Failed to write trusted root xxxx to backup file.	**Cause:** The trusted root identified in error messages did not get saved to the backup file. **Action:** An exception string included in the message might provide additional information.

Event Code	*Description*	Remedy
201003046	Failed to retrieve trusted root xxxx from eDirectory.	**Cause:** The certtool failed to retrieve the trusted root identified in the error. Likely a PKI or eDirectory error. **Action:** This error will be accompanied by an error string.
201003048	Not all items were backed up.	**Cause:** See accompanying errors. **Action:** Refer to previous error messages to identify which certificates or trusted roots were not backed up.
201003049	Failed to retrieve the CA xxxx from eDirectory. Likely a PKI or eDirectory error.	**Cause:** The certtool failed to retrieve the CA key identified in the error. **Action:** This error will be accompanied by an error string.
201003050	Failed to write CA key xxxx to backup file.	**Cause:** The CA key identified in the error did not get written to the backup file. **Action:** An exception string included in the message my provide additional information.
201003051	Failed to open backup file for reading.	**Action:** Make sure the backup file exists. Do not include `.ldif` or `.zip` in the name of the back up file. **Action:** Make sure the user logged in has sufficient rights to access the file.
201003052	Not all items were restored.	**Cause:** See accompanying errors. **Action:** Refer to previous error messages to identify which certificates or trusted roots were not backed up.
301003053	Error closing backup.	**Action:** This error occurred after all restore operations had completed. Should not cause any problem.
201003056	Error importing CA key: xxxx	**Action:** The CA key was not restored. See the accompanying Error for more information. Likely a PKI error. **Action:** Make sure the password you provided matches the encryption password used when backing up the data.
201003057	Error importing key: xxxx	**Cause:** The CA key was not restored. See the accompanying Error for more information. Likely a PKI error. **Action:** Make sure the password you provided matches the encryption password used when backing up the data.
201003058	Error importing trusted root: xxxx	**Cause:** The trusted root was not restored. See the accompanying Error for more information. Likely a PKI error.
	VCDN configuration	

Event Code	*Description*	**Remedy**
201004001	Failed to configure VCDN objects for data store access.	The VCDN user objects were not restored with their passwords. Device Manager will not start up until the passwords have been properly set. **Cause:** The vcdnbkup.jar utility failed to reset passwords for VCDN objects. This causes errors starting up device manager. **Action:** Make sure `/opt/volera/roma/conf/vcdn.conf` file is present and has the correct information. To fix enter the following command in the /opt/novell/devman/bin directory: `java -jar vcdnbkup.jar -userid cn=admin,o=novell -pwd admin_password -vcdnUser`
201004002	Application Error.	The VCDN user objects were not restored with their passwords. Device Manager will not start up until the passwords have been properly set. Accompanied by a stack trace with more information. **Cause:** vcdnbkup.jar utility failed to reset passwords for VCDN objects. This will cause errors starting up device manager. **Action:** Make sure the information in `/opt/volera/roma/conf/vcdn.conf` file is correct: Fix the file by running the following command (in /opt/novell/devman/bin): `java -jar vcdnbkup.jar -userid cn=admin,o=novell -pwd admin_password -vcdnUser`

NetIQ Modular Authentication Class (012)

The NetIQ Modular Authentication Service (NMAS) Class provides access to a number of advanced authentication mechanisms available from Novell, Inc. and Novell partners.

Component 012

- Subgroup 01: General/Configuration
- Log file: catalina.out for trace and application level logging as enabled by the log settings (click **Identity Server** > **Edit** > **Logging)**

Event Code	Description	Remedy
General/Configuration		

Event Code	Description	Remedy
301201001	NMAS Authentication Class	The log message language resource file could not be located. **Cause:** The log message language resource file was not found **Action:** Verify installation.
101201002	NMAS Authentication Class	Error getting LDAP host address. **Cause:** System configuration. **Action:** Verify installation and availability of LDAP host server.
101201003	NMAS Authentication Class	The NMAS_LOGIN_SEQUENCE initialization property were not provided. **Cause:** The NMAS_LOGIN_SEQUENCE property was not defined for the authentication class. **Action:** Use the management interface to add the NMAS_LOGIN_SEQUENCE property to either the class or the method, and assign it the name of a valid NMAS login sequence.
101201004	NMAS Authentication Class	Unable to write to HTTPResponse **Cause:** Unknown **Action:** Check system status.
501201005	NMAS Authentication Class	UserID not found. **Cause:** Invalid User ID. **Action:** Verify username
101201006	NMAS Authentication Class	Invalid NMAS Login state. **Cause:** Unknown **Action:** Check server status.
101201007	NMAS Authentication Class	NMAS Login Error. **Cause:** See NMAS Error codes. **Action:** Indicated by NMAS error code.

IV Appendix

The following sections contain additional documentation and information about Access Manager:

- Appendix A, “Data Model Extension XML,” on page 1187
- Appendix B, “SOAP versus REST API,” on page 1193
- Appendix C, “OAuth versus Other Protocols,” on page 1195
- Appendix D, “Access Manager Reports Samples,” on page 1197

A Data Model Extension XML

The data model for some Web services is extensible. You can enter XML definitions of data model extensions in a custom profile (for more information, see “Modifying Service and Profile Details for Employee, Custom, and Personal Profiles” on page 464). Data model extensions hook into the existing Web service data model at predefined locations.

All schema model extensions reside inside of a schema model extension group. The group exists to bind model data items together under a single localized group name and description. Schema model extension groups can reside inside of a schema model extension root or inside of a schema model extension. There can only be one group per root or extension. Each root is hooked into the existing Web service data model. Multiple roots can be hooked into the same location in the existing Web service data model. This conceptual model applies to the structure of the XML that is required to define data model extensions.

The high-level view of the data model extension XML is as follows:

```
<SchemaExtensions>
   <Root>
     <Group>
       <Extension>
          <Group>
           <Extension>...</Extension>
           <Extension>...</Extension>
           ...
          </Group>
       </Extension>
       <Extension>
         <ValueSet>
            <Value/>
            <Value/>
         </ValueSet>
       </Extension>
      ...
    </Group>
  <Root>
<Root>...</Root>
...
</SchemaExtensions>
```

Elements

The definition of the attributes for each data model extension XML element are as follows:

- “Root Element” on page 1188
- “Group Element” on page 1188
- “Extension Element” on page 1188
- “ValueSet Element” on page 1189
- “Value Element” on page 1190

Root Element

parent: The unique identifier of the “hook point” in the Web service’s data model. These hook points are defined by the Web service data model schema. These unique identifiers represent the xpaths of each data item within the model schema. Possible values for the parent attribute are listed in Table A-1:

Table A-1 *Root Element*

Personal Profile	/pp:PP/pp:Extension /pp:PP/pp:CommonName/pp:Extension /pp:PP/pp:CommonName/pp:AnalyzedName/pp:Extension /pp:PP/pp:LegalIdentity/pp:Extension /pp:PP/pp:LegalIdentity/pp:VAT/pp:Extension /pp:PP/pp:LegalIdentity/pp:AltID/pp:Extension /pp:PP/pp:EmploymentIdentity/pp:Extension /pp:PP/pp:AddressCard/pp:Extension /pp:PP/pp:AddressCard/pp:Address/pp:Extension /pp:PP/pp:MsgContact/pp:Extension /pp:PP/pp:Facade/pp:Extension /pp:PP/pp:Demographics/pp:Extension
Employee Profile	/ep:EP/ep:Extension /ep:EP/ep:CorpCommonName/ep:Extension /ep:EP/epCorpLegalIdentity/ep:Extension /ep:EP/ep:CorpLegalIdentity/ep:VAT/ep:Extension /ep:EP/ep:CorpLegalIdentity/ep:AltID/ep:Extension
Open Profile	/op:OP/op:Extension /op:OP/op:CustomizableStringsop:Extension

package (required): The Java package name where all classes for this root are implemented. This includes resource description classes and data model instance classes. For example, com.novell.nids.profile.model.extensions.

resourceClass (required): The Java class name of the resource description class that is used to load all resources associated with this root. Because resource description class files are assumed to reside in the root’s package, only the filename is needed. Resource description classes are Java classes that must be created by the person extending the model. You must also extend the com.novell.nidp.resource.NIDPResDesc class.

Group Element

resourceID: The resource ID of the display name of the group. This resource ID is assumed to be a key in the resource bundle supplied by the resource description class file associated with the containing root.

descriptionResourceID: The resource ID of the description of the group. This resource ID is assumed to be a key in the resource bundle supplied by the resource description class file associated with the containing root.

Extension Element

name (required): The name of the data model extension. This name must be the name of the XML element that will be used in the data model.

class (optional): The Java class name of the data model instance class. Because data model instance class files are assumed to reside in the root's package, only the filename is needed. If this attribute is omitted, then the value of the name attribute must be the instance class filename.

syntax: The syntax of this data model extension. Possible values are:

- String
- LocalizedString
- Container

format: Required if the syntax is *String* or *LocalizedString*. The syntax of this data model extension. Possible values are:

- CaseIgnore
- CaseExtract
- URI
- URL
- Date
- DateNoYear
- CountryCode
- LanguageCode
- KeyInfo
- Number

upper: The upper bound of a numeric value. Use this attribute only if the format attribute value is Number. The value is a signed integer. If this attribute is omitted, the default value is java.lang.Integer.MAX_VALUE.

lower (optional): The lower bound of a numeric value. This attribute is only used if the format attribute value is Number. The value is a signed integer. If this attribute is omitted, the default value is java.lang.Integer.MIN_VALUE.

min (required): The cardinality of the XML element represented by this data model extension. It is the minimum number of elements allowed. The value is an unsigned integer. If this attribute is omitted, the default value is 0.

max (required): The cardinality of the XML element represented by this data model extension. It is the maximum number of elements allowed. The value is an unsigned integer. If this attribute is omitted, the default value is 1. The value UNBOUNDED may be used to indicate that there are no bounds.

namingClass: (required if syntax equals Container and max is UNBOUNDED). The class that is used as the naming attribute for the container. The class must represent one of the immediate children of the container. This class is used to name each instance of the container.

ValueSet Element

A ValueSet element contains a set of fixed values that a data model entry can contain. If a data model extension has a ValueSet, the user interface to edit the value of that extension limits the user to these values. The ValueSet element has no attributes.

Value Element

A Value element represents a value in a ValueSet. It contains the actual value to be stored in the data model entry and the display name resource ID associated with the value.

resourceID (required): The resource ID of the display name of the value. This resource ID is assumed to be a key in the resource bundle supplied by the resource description class file associated with the containing root.

value (required): The value stored in the data model entry.

name (required): The name of the data model extension. This name must be the name of the XML element that is used in the data model.

Writing Data Model Extension XML

Data model extension XML must be defined in the namespace novell:liberty:wsf:config:1:0:0 and that namespace must be defined on the SchemaExtensions element. Normally, the namespace prefix wsfc is used. An example of data model extension XML is:

```
<wsfc:SchemaExtensions xmlns:wsfc="novell:liberty:wsf:config:1:0:0">
  <wsfc:Root parent="/pp:PP/pp:Facade/pp:Extension"
     package="com.novell.nidp.liberty.wsf.idsis.ppservice.extensions"
     resourceClass="PPExtensionsResDesc">
   <wsfc:Group resourceId="PP.EXT.FC.GROUP"
      descriptionResourceId="PP.EXT.FC.GROUP.DESC">
   <wsfc:Extension name="AliasName"
       class="FacadeAliasName"
       syntax="String"
       format="CaseIgnore"
       resourceId="PP.EXT.FC.AliasName"
       min="0" max="1"/>
   <wsfc:Extension name="FavoriteURLs"
       class="FacadeFavoriteURLs"
       syntax="String"
       format="CaseExact"
       resourceId="PP.EXT.FC.FavoriteURLs" min="0" max="UNBOUNDED"/>
   </wsfc:Group> </wsfc:Root>
   <wsfc:Root parent="/pp:PP/pp:Demographics/pp:Extension"
      package="com.novell.nidp.liberty.wsf.idsis.ppservice.extensions"
      resourceClass="PPExtensionsResDesc">
   <wsfc:Group resourceId="PP.EXT.DM.GROUP"
      descriptionResourceId="PP.EXT.DM.GROUP.DESC">
   <wsfc:Extension name="EyeColor"
      class="DemographicsEyeColor"
      syntax="String" format="URI"
      resourceId="PP.EXT.DM.EyeColor"
      min="0"
      max="UNBOUNDED">
   <wsfc:ValueSet>
<wsfc:Value resourceId="PP.EXT.DM.HC.Blue" value="urn:pp:dm:blue"/>
<wsfc:Value resourceId="PP.EXT.DM.HC.Brown" value="urn:pp:dm:brown"/>
<wsfc:Value resourceId="PP.EXT.DM.HC.Green" value="urn:pp:dm:green"/>
<wsfc:Value resourceId="PP.EXT.DM.HC.Gray" value="urn:pp:dm:gray"/>
<wsfc:Value resourceId="PP.EXT.DM.HC.Hazel" value="urn:pp:dm:hazel"/>
</wsfc:ValueSet>
</wsfc:Extension>
</wsfc:Group>
</wsfc:Root>
```

```
<wsfc:Root parent="/pp:PP/pp:Extension"
   package="com.novell.nidp.liberty.wsf.idsis.ppservice.extensions"
   resourceClass="PPExtensionsResDesc">
<wsfc:Group resourceId="PP.EXT.AU.GROUP"
  descriptionResourceId="PP.EXT.AU.GROUP.DESC">
<wsfc:Extension name="Automobile"
  class="Automobile"
  syntax="Container"
  resourceId="PP.EXT.Automobile"
  min="0"
  max="UNBOUNDED"
  namingClass="AutomobileLicensePlate">
<wsfc:Group resourceId="PP.EXT.AU.DETAILS.GROUP"
  descriptionResourceId="PP.EXT.AU.DETAILS.GROUP.DESC">
<wsfc:Extension name="AutomobileModel"
  class="AutomobileModel"
  syntax="String"
  resourceId="PP.EXT.AU.Model"
  min="0"
  max="1"/>
<wsfc:Extension name="AutomobileMake"
  class="AutomobileMake"
  syntax="String"
  format="CaseIgnore"
  resourceId="PP.EXT.AU.Make"
  min="0"
  max="1"/>
<wsfc:Extension name="AutomobileLicensePlate"
  class="AutomobileLicensePlate"
  syntax="String"
  format="CaseIgnore"
  resourceId="PP.EXT.AU.LicensePlate"
  min="0" max="1"/>
</wsfc:Group>
</wsfc:Extension>
</wsfc:Group>
</wsfc:Root>
</wsfc:SchemaExtensions>
```

B SOAP versus REST API

The following table compares SOAP with REST:

SOAP	REST
Stands for Simple Object Access Protocol	Stands for Representational State Transfer
An XML based message protocol	Does not enforce message format
Follows stateful implementation	Follows stateless model
No error handling	Built-in error handling
Strongly typed, strict specification for implementation	Less restrictive about the implementation
Uses interfaces and named operations to expose business logic	Uses URI and methods to expose resources
Both SMTP and HTTP are valid application layer protocols used as Transport for SOAP	Tied to the HTTP transport model
More verbose	Less verbose
Uses WSDL for communication between consumer and provider	Uses XML or JSON to send and receive data
Invokes services by calling RPC method	Invokes services through URL path

C OAuth versus Other Protocols

The following table lists the differences among OAuth, OpenID Connect, WS-Trust, WS Fed, and SAML:

Table C-1 *Differences among OAuth, OpenID Connect, and WS*-Family*

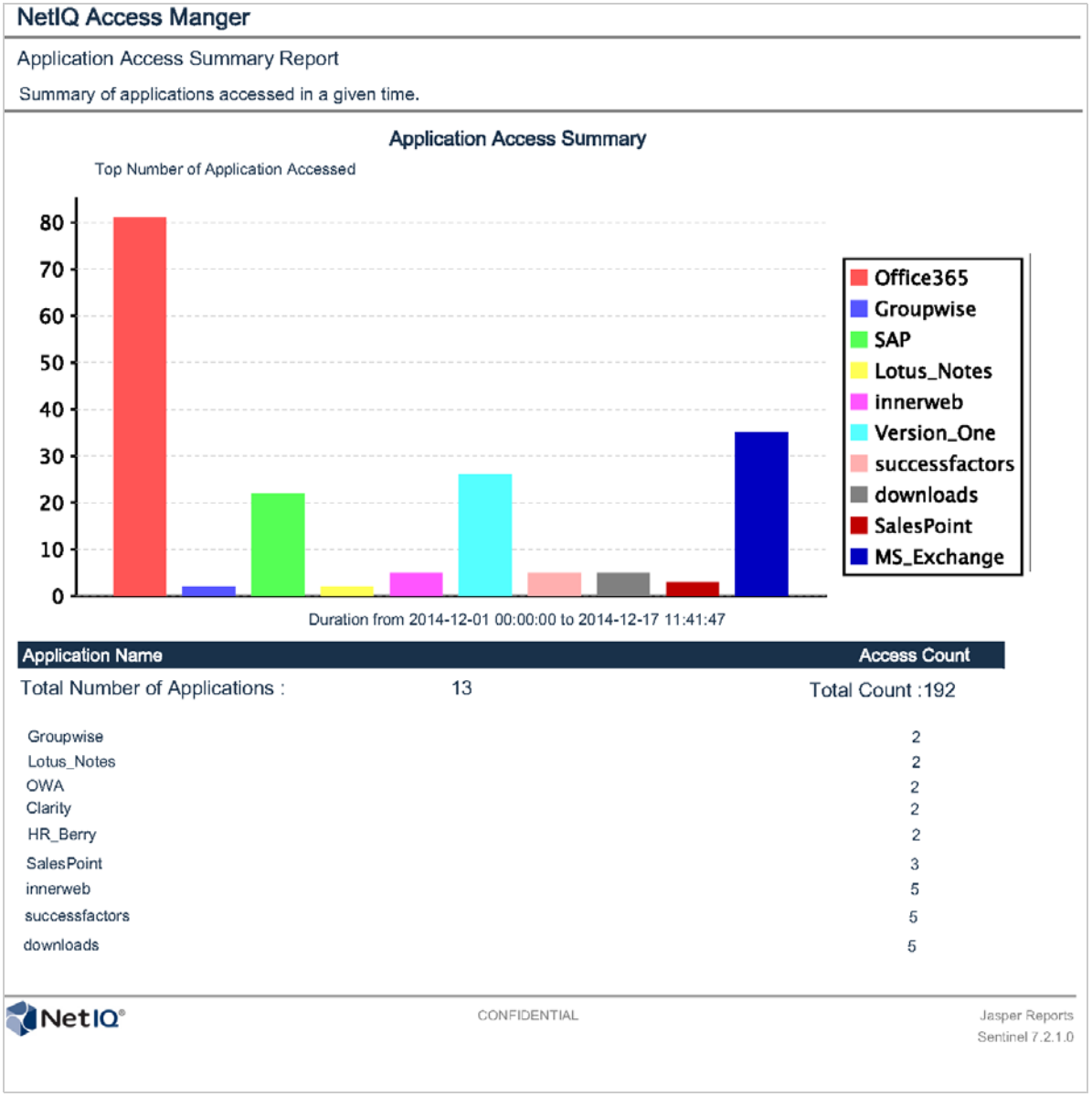
OAuth	OpenID Connect	SAML	WS-* Family
An open protocol to allow secure authorization in a simple and standard method from web, mobile and desktop applications. Provides API authorization between applications.	Provides single sign-on (SSO) layer on top of the OAuth protocol for consumers.	An XML-based open standard data format for exchanging authentication and authorization data between an identity provider and a service provider. Encompasses profiles, bindings and constructs to achieve SSO, federation, and identity management.	Allows secure identity propagation and token exchange between Web services. Enables applications to construct trusted SOAP message exchanges.
OAuth tokens can be binary, JSON, or SAML	Uses JSON tokens	Deals with XML as the data construct or token format.	Uses Request Security Token (RST) and Request Security Token Response (RSTR)
Uses HTTP exclusively	Uses HTTP exclusively	No restriction on the transport format. You can use SOAP, JMS, or any transport you want to use to send SAML tokens or messages.	No restriction on the transport format. You can use SOAP, JMS, or any transport you want to use to send SAML tokens or messages.
Designed for use with applications on the Internet.	Designed for use with applications on the Internet.	Used in enterprise SSO scenarios.	Used in enterprise SSO scenarios.

D Access Manager Reports Samples

This section provides samples of reports that you can generate by using the Access Manager Reporting Solution Pack.

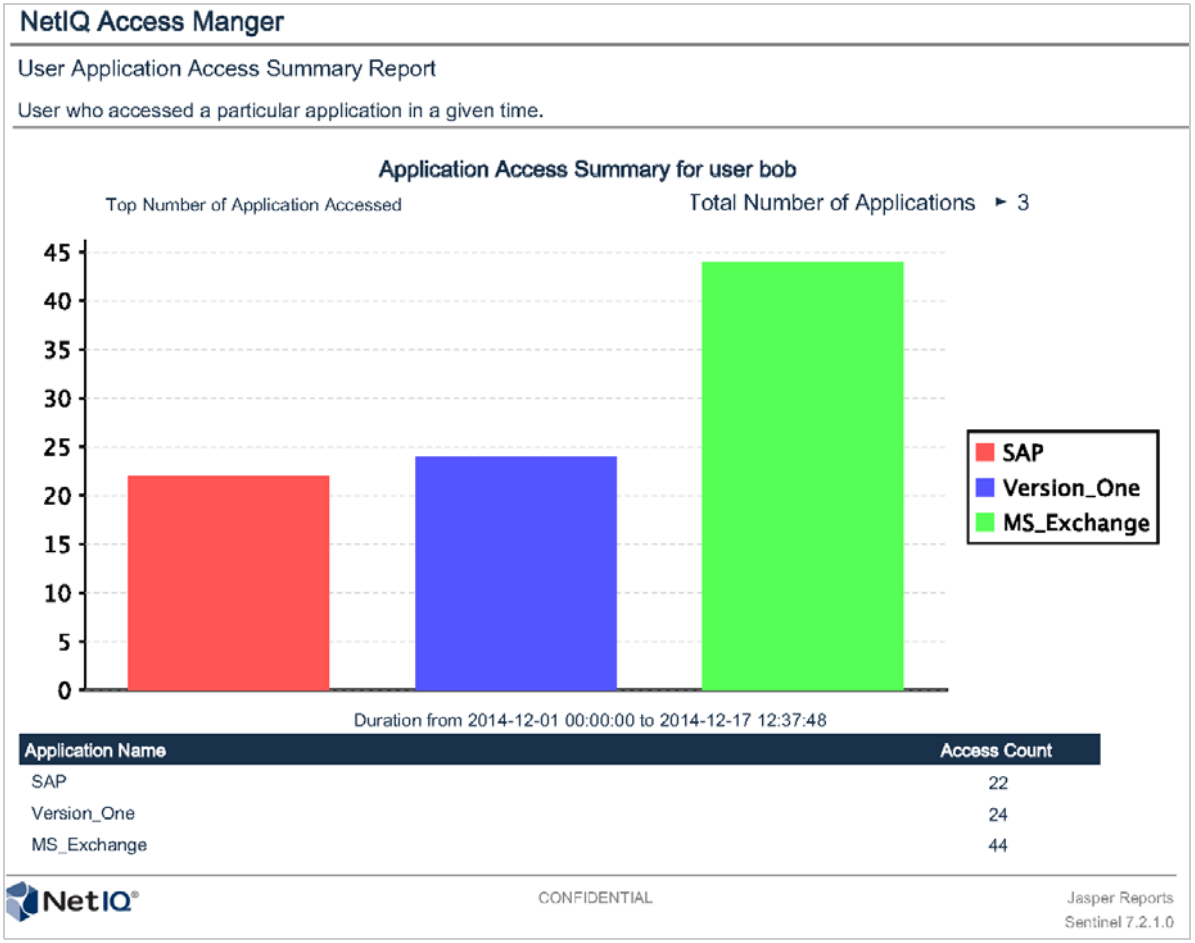
- “Application Access Summary Report” on page 1198
- “User Application Access Summary Report” on page 1199
- “Application Specific User Access Report” on page 1200
- “Federation Summary Report” on page 1201
- “User Login Contract Summary Report” on page 1202
- “User Login Failure Report” on page 1203
- “Application Specific Risk based Authentication Report” on page 1204

Application Access Summary Report

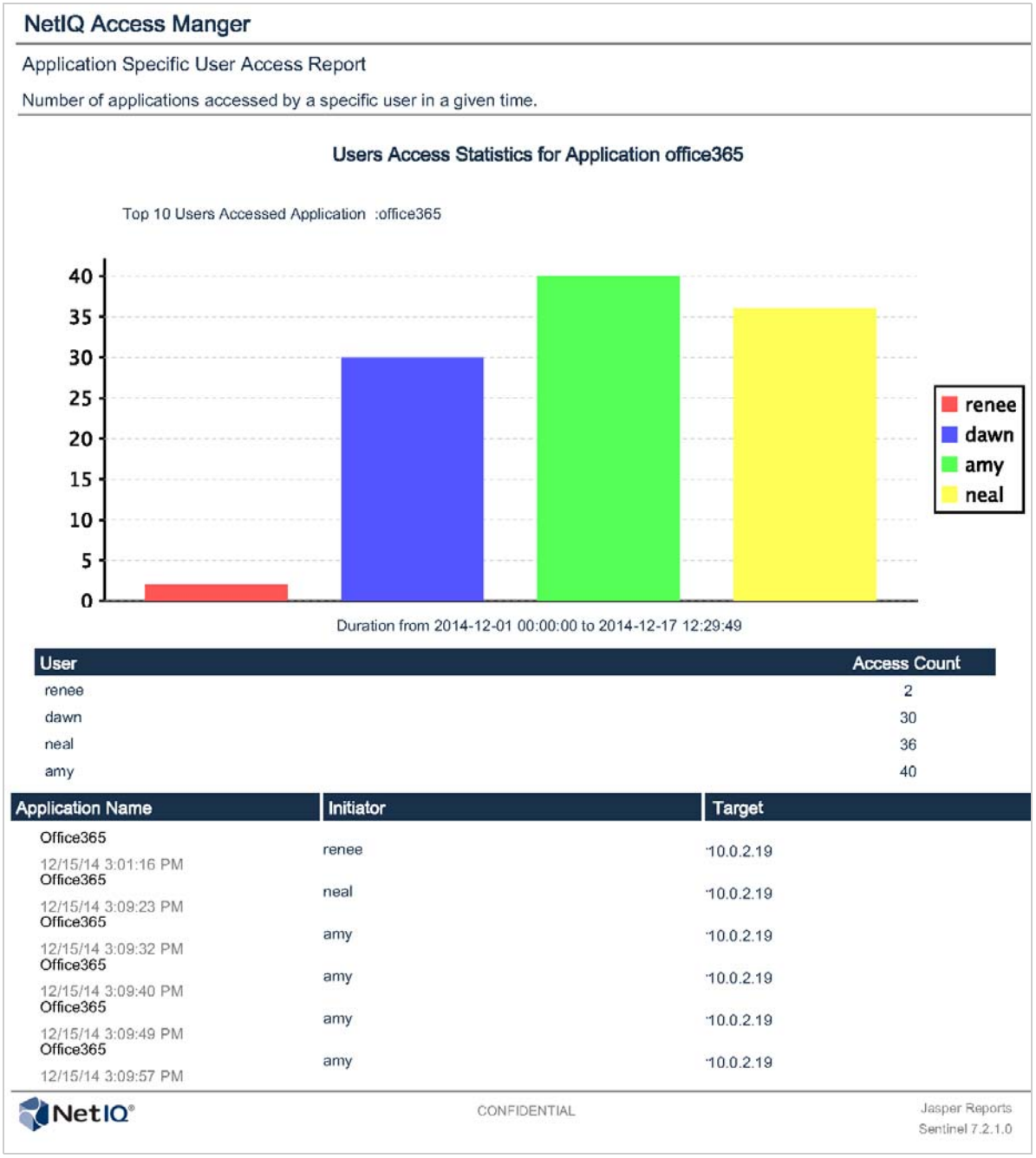


NOTE: In this sample report, Office365 and SuccessFactors are names of the protected resources configured in the Access Gateway reverse proxies. These are not federated Office 365 and Success Factor requests.

User Application Access Summary Report



Application Specific User Access Report



NetIQ Access Manger

Application Specific User Access Report

Number of applications accessed by a specific user in a given time.

Users Access Statistics for Application office365

Top 10 Users Accessed Application :office365

Duration from 2014-12-01 00:00:00 to 2014-12-17 12:29:49

User	Access Count
renee	2
dawn	30
neal	36
amy	40

Application Name	Initiator	Target
Office365 12/15/14 3:01:16 PM	renee	10.0.2.19
Office365 12/15/14 3:09:23 PM	neal	10.0.2.19
Office365 12/15/14 3:09:32 PM	amy	10.0.2.19
Office365 12/15/14 3:09:40 PM	amy	10.0.2.19
Office365 12/15/14 3:09:49 PM	amy	10.0.2.19
Office365 12/15/14 3:09:57 PM	amy	10.0.2.19

NetIQ CONFIDENTIAL Jasper Reports Sentinel 7.2.1.0

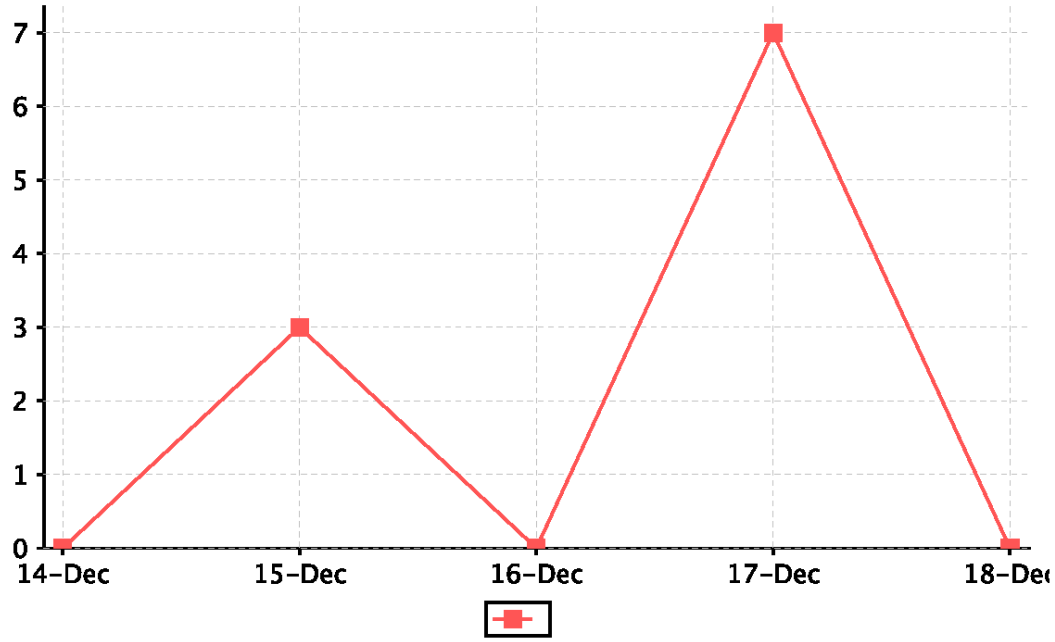
Federation Summary Report

NetIQ Access Manger

Federation Summary

Users who accessed a federated service in a given time.

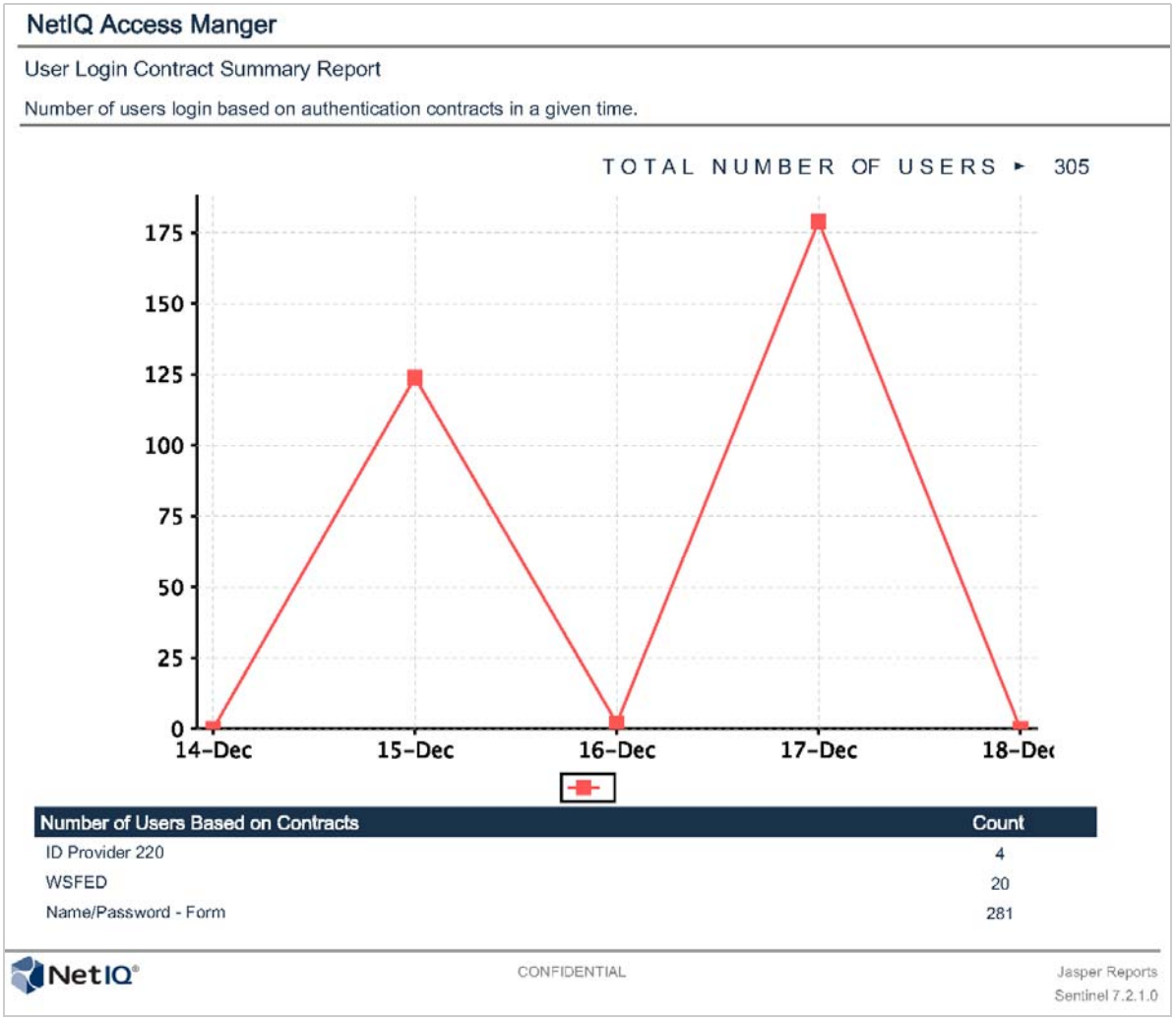
TOTAL FEDERATED USERS FROM IDENTITY PROVIDERS AND SERVICE PROVIDERS ► 10



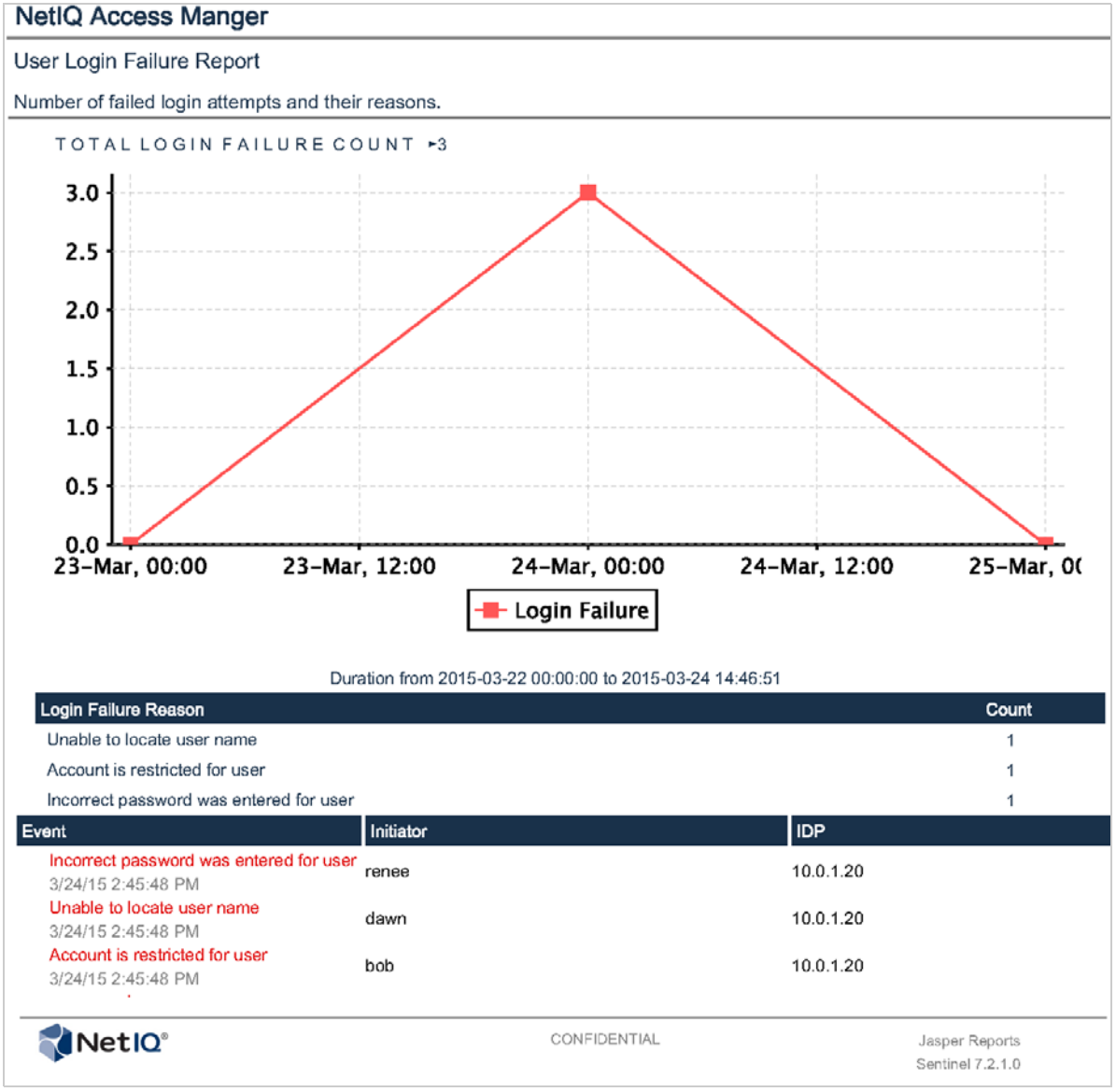
Duration from 2014-12-01 00:00:00 to 2014-12-17 12:42:52

List of Identity Providers and Service Providers	Count
ID Provider 220	3
Service Provider 99	7

User Login Contract Summary Report



User Login Failure Report



NetIQ Access Manger

User Login Failure Report

Number of failed login attempts and their reasons.

TOTAL LOGIN FAILURE COUNT =3

Duration from 2015-03-22 00:00:00 to 2015-03-24 14:46:51

Login Failure Reason	Count
Unable to locate user name	1
Account is restricted for user	1
Incorrect password was entered for user	1

Event	Initiator	IDP
Incorrect password was entered for user 3/24/15 2:45:48 PM	renee	10.0.1.20
Unable to locate user name 3/24/15 2:45:48 PM	dawn	10.0.1.20
Account is restricted for user 3/24/15 2:45:48 PM	bob	10.0.1.20

NetIQ

CONFIDENTIAL

Jasper Reports
Sentinel 7.2.1.0

Application Specific Risk based Authentication Report



NetIQ Sentinel Report as run on November 20, 2015 at 4:36:45 PM IST Page 1 / Total 2

NetIQ Access Manager

Application Specific Risk based Authentication Report

Summary of Risk Based Authentication for particular application in a given time.

Duration from 2015-11-19 00:00:00 to 2015-11-20 16:36:39

Risk Level	Access Count
High	8
Medium	11
Low	14

Initiator	Risk Score/ Risk Level	Action
cn=micky,o=\novell	20	
11/20/15 10:50:36 AM	Medium	STEP_UP/Additional authentication
cn=julia,o=\novell	0	
11/20/15 10:49:13 AM	Low	Deny On Rule Pass Executed.DENY
cn=alice,o=\novell	20	
11/20/15 10:45:42 AM	Low	ALLOW
cn=bob,o=\novell	20	
11/20/15 10:45:13 AM	Low	ALLOW
cn=alice,o=\novell	20	
11/19/15 9:53:00 PM	Low	ALLOW
cn=alice,o=\novell	0	
11/19/15 9:52:21 PM	Low	Deny On Rule Pass Executed.DENY
cn=alice,o=\novell	0	
11/19/15 9:50:35 PM	Low	Deny On Rule Pass Executed.DENY
cn=alice,o=\novell	20	
11/19/15 9:50:12 PM	Medium	STEP_UP/Additional authentication
cn=alice,o=\novell	20	
11/19/15 9:49:02 PM	Medium	STEP_UP/Additional authentication

NetIQ CONFIDENTIAL Sentinel 7.3.1.0