CloudAccess

Connectors Guide

February 2017

MICRO FOCUS®

Legal Notice

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

Contents

About this Book and the Library 7

About NetIQ Corporation 9

1 Overview of CloudAccess Connectors 11

1.1 Understanding Single Sign-On Methods . . . 11
1.1.1 Federated Single Sign-On with SAML 2.0 or WS-Federation . . . 11
1.1.2 Basic Single Sign-On. . . . 12
1.1.3 OAuth 2.0 Single Sign-On . . . 12
1.1.4 Bookmarks. . . . 13
1.2 Connectors for Federated Single Sign-On and Provisioning . . . 13
1.3 Connectors for Federated Single Sign-On . . . 14
1.4 Connectors for Basic Single Sign-On. . . . 14
1.5 Connector for OAuth 2.0 Single Sign-On . . . 15
1.6 Connector for Bookmarks . . . 15
1.7 Custom Connectors . . . 15

2 Configuring Connectors 17

2.1 Requirements for Connectors . . . 17
2.2 Viewing Connectors for Applications . . . 17
2.3 Providing Access to Applications for Users . . . 18
2.4 How CloudAccess Provisions User Accounts . . . 18
2.4.1 Requirements for Provisioning. . . . 18
2.4.2 Understanding Provisioning. . . . 19
2.4.3 Samples of Account Creations . . . 20
2.4.4 CloudAccess Default Naming Convention for Newly Provisioned Accounts . . . 21
2.4.5 Matching Criteria for Merging Existing Accounts . . . 21
2.4.6 Understanding Naming Policy Options . . . 24
2.4.7 Understanding Collision Handling in Provisioning . . . 25

3 Creating Custom Connectors 27

3.1 Accessing the Access Connector Toolkit . . . 27
3.2 Toolkit Requirements . . . 28
3.2.1 Toolkit Compatibility . . . 28
3.2.2 Provisioning Support . . . 28
3.3 Creating a SAML 2.0 Connector Template. . . . 28
3.3.1 SAML 2.0 Requirements for the Application Service Provider. . . . 28
3.3.2 Planning for a SAML 2.0 Connector . . . 29
3.3.3 Creating a SAML 2.0 Connector Template for an Application . . . 29
3.4 Creating a WS-Federation Connector Template . . . 30
3.4.1 WS-Federation Requirements for the Application Service Provider . . . 30
3.4.2 Planning for a WS-Federation Connector . . . 30
3.4.3 Creating a WS-Federation Connector Template for an Application . . . 31
3.5 Creating a SAML 2.0 Inbound (SAML2 In) Connector Template . . . 31
3.5.1 Understanding SAML2 In Identity Sources . . . 32
3.5.2 Requirements for Using SAML2 In Identity Sources . . . 33
3.5.3 SAML2 In Requirements for the Identity Provider . . . 34
3.5.4 Planning for a SAML2 In Connector . . . 34
3.5.5 Creating a SAML2 In Connector for an Identity Provider. . . . 35

3.6 Creating a Basic SSO Connector Template 35
3.6.1 Basic SSO Requirements 35
3.6.2 Planning for Basic SSO 36
3.6.3 Creating a Basic SSO Connector Template for a Web Service 36
3.7 Modifying a Connector 37
3.8 Exporting a Connector Template 38
3.9 Importing and Configuring Custom Connectors 38
3.9.1 SAML2 and WS-Fed Custom Connectors 38
3.9.2 SAML2 In Custom Connectors 39
3.9.3 Basic SSO Custom Connectors. 40

4 Importing Connectors from the Application Connector Catalog 43

5 Configuring Appmarks for Connectors 45

5.1 Understanding Appmarks. 45
5.2 Configuring an Appmark for the Desktop Browser or Mobile Device 47
5.3 Creating Multiple Appmarks for an Application 48
5.4 Using Appmark Variables 49
5.5 Policy Mapping for Non-Public Appmarks 49

6 Connector for Bookmarks 51

6.1 Configuring the Connector for Bookmarks 51

7 Connector for OAuth2 Resources 53

7.1 Configuring the OAuth2 Client Application 53
7.2 Configuring the Connector for OAuth2 Resources. 54
7.3 Supported OpenID Connect Schema. 55

8 Connectors for Basic SSO 57

8.1 Requirements for Using Basic SSO with Applications 57
8.2 Understanding Basic Single Sign-On. 58
8.3 Importing and Configuring a Connector for Basic SSO 60
8.4 Troubleshooting Basic Single Sign-On. 61

9 Connector for Google Apps 63

9.1 Connector Requirements 63
9.2 Understanding Google Apps Provisioning 64
9.3 Creating a Google Service Account 64
9.4 Configuring the Connector for Google Apps 65
9.5 Enabling Google Mail Proxy. 66
9.6 Configuring Multiple Connectors for Google Apps 67

10 Connector for Microsoft Office 365 69

10.1 How the Connector for Office 365 Works. 69
10.1.1 Setup and Configuration 70
10.1.2 User Provisioning. 71
10.1.3 User Login to Office 365 71
10.2 Connector Requirements 72
10.3 Installing the Connector for Office 365. 73

10.4 Configuring the Connector for Office 365 75
10.5 Validating the Connector for Office 365 Installation 75
10.6 Configuring Appmarks for Office 365 Applications 76
10.7 Changing the Configuration of the Connector 76
10.8 Changing the Name of an Office 365 Security Group 76
10.9 Upgrading the Connector from SAML 2.0 to WS-Federation 76
10.10 Uninstalling the Connector for Office 365 77
10.11 Installing Multiple Connectors for Office 365 77

11 Connector for Salesforce **79**

11.1 Connector Requirements 79
11.2 Configuring Salesforce to Trust CloudAccess 80
11.3 Configuring the Connector for Salesforce 80
11.4 Configuring Appmarks for Salesforce 82
11.5 Configuring Multiple Connectors for Salesforce 83
11.6 Using SSO to Salesforce on Mobile Devices 83
11.6.1 Understanding the Mobile SSO Process 84
11.6.2 Requirements for Mobile SSO to Salesforce 85
11.7 Configuring Delegated Authentication in Salesforce 86
11.8 Configuring the Salesforce Federation Identifier 87

12 Connector for ServiceNow **89**

12.1 Connector Requirements 89
12.2 Configuring the Connector for ServiceNow 90
12.3 Configuring Appmarks for ServiceNow 91

13 Single Sign-On Connectors **93**

13.1 Global Requirements for the Single Sign-On Connectors 93
13.2 Requirements for the Single Sign-On Connectors 93
13.3 Configuring Single Sign-On Connectors 94
13.4 Configuring ADFS to Connect to SharePoint 95
13.4.1 Requirements. 95
13.4.2 Modifying the Connector for ADFS Template 95
13.4.3 Importing the Modified Connector 96
13.4.4 Configuring the Modified Connector 97
13.4.5 Modifying Claim Rules in the ADFS System 97
13.4.6 Configuring ADFS to Send SharePoint the Claim Rules 97
13.4.7 Configuring People Picker to Specify the Roles 98

14 Troubleshooting CloudAccess Connectors **99**

14.1 Using Troubleshooting Tools for Application Access Issues 99
14.2 Troubleshooting Connector States 100
14.3 Troubleshooting Provisioning Issues 101
14.4 Troubleshooting Google Apps Issues 102
14.4.1 Google Apps Users Can No Longer Log In After Enabling Single Sign-On. 102
14.4.2 Users Are Not Provisioned to the Correct Organization 103
14.4.3 Chrome Profiles Cause Logins to Fail 103
14.4.4 Users Who Are Provisioned to Multiple Google Domains Cannot Access Original Mailbox 103
14.5 Troubleshooting Salesforce Issues 103
14.5.1 Salesforce Login Issues. 104

14.5.2 Behavior of Service Provider-Initiated Login To Salesforce When Kerberos Is Enabled 104
14.6 Troubleshooting Office 365 Issues 105
14.6.1 Obtaining Installation and Provisioning Logs 105
14.6.2 Office 365 Logout Error on Mobile Devices 105
14.6.3 Display Name Does Not Change in Office 365 after Changing It in Identity Source 105
14.6.4 Office Web Apps Cannot Be Assigned or Unassigned Without SharePoint Online. 106
14.6.5 Connectors for Office 365 that are Configured for Domain and Subdomains Do Not Work Correctly 106
14.6.6 Installation Fails with Error: System.Runtime.InteropServices.COMException 106
14.6.7 Renaming an Authorization for an Office 365 Account Requires Policy Remapping in CloudAccess 106
14.7 Troubleshooting ServiceNow Issues 107
14.8 Troubleshooting Failed Logins with Basic SSO Connectors 107
14.9 Troubleshooting Other Connector Issues. 107
14.9.1 Logging Out of Identity Provider Landing Page Does Not Result in Logging Out of SaaS Accounts 107
14.9.2 Admin Page Does Not Provide a Way to View SaaS Metadata 107
14.9.3 User Session Timeout Affects Box Connector 108
14.10 Troubleshooting Custom Connectors 108

A Custom Connector Worksheets 109

A.1 Worksheet for SAML or WS-Federation Custom Connectors 109
A.2 Worksheet for SAML In Custom Connectors 110
A.3 Worksheet for Basic SSO Custom Connectors 111

About this Book and the Library

The *CloudAccess Connectors Guide* provides information on importing, configuring, and managing the connectors that you use with CloudAccess.

Intended Audience

This guide provides information for CloudAccess administrators who are responsible for configuring and managing the connectors used with CloudAccess.

Other Information in the Library

The library provides the following information resources:

Installation and Configuration Guide

Provides installation and configuration instructions for CloudAccess.

Help

Provides context-sensitive information and step-by-step guidance for common tasks.

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

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
Website:	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
Website:	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 website in HTML and PDF formats on a page that does not require you to log in. If you have suggestions for documentation improvements, click **Add Comment** 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 community.netiq.com.

1 Overview of CloudAccess Connectors

CloudAccess uses connectors to provide single sign-on (SSO) access for users to web resources through CloudAccess. CloudAccess authenticates the users against your identity sources. When the user accesses the link for an application through CloudAccess, CloudAccess shares the authenticated user's identity information with the destination application to establish the user's session. Each user can access only the links they are authorized to use, according to the entitlements you set for each application.

- Section 1.1, "Understanding Single Sign-On Methods," on page 11
- Section 1.2, "Connectors for Federated Single Sign-On and Provisioning," on page 13
- Section 1.3, "Connectors for Federated Single Sign-On," on page 14
- Section 1.4, "Connectors for Basic Single Sign-On," on page 14
- Section 1.5, "Connector for OAuth 2.0 Single Sign-On," on page 15
- Section 1.6, "Connector for Bookmarks," on page 15
- Section 1.7, "Custom Connectors," on page 15

1.1 Understanding Single Sign-On Methods

CloudAccess supports single sign-on for a variety of web services and applications that have different authentication requirements. The method used for single sign-on depends on the security requirements and capabilities of each destination resource.

- Section 1.1.1, "Federated Single Sign-On with SAML 2.0 or WS-Federation," on page 11
- Section 1.1.2, "Basic Single Sign-On," on page 12
- Section 1.1.3, "OAuth 2.0 Single Sign-On," on page 12
- Section 1.1.4, "Bookmarks," on page 13

1.1.1 Federated Single Sign-On with SAML 2.0 or WS-Federation

Federated single sign-on relies on a trust relationship between an identity provider and a service provider to give a user access to a protected web service or application through CloudAccess. Open standards for federation include SAML 2.0 (Security Assertion Markup Language), WS-Federation (Web Services Federation), and SAML 2.0 Inbound. They provide a vendor-neutral means of exchanging user identity, authentication, and attribute information. The service provider trusts the identity provider to validate the user's authentication credentials and to send identity information about the authenticated user. The service provider accepts the data and uses it to give the user access to the destination service or application. This data exchange is transparent for the user. It allows the user to access the web service or application without providing an additional password.

The following describes the SSO experience for trusted access to an application through CloudAccess:

1. Users provide login credentials directly to CloudAccess, such as their corporate user name and password.
2. CloudAccess authenticates the user's credentials against the identity sources.

3. CloudAccess presents the landing page to the user with links to applications that the user is entitled to use.
4. When a user clicks an application’s link, as the identity provider, CloudAccess produces an authentication assertion or token for the service provider that contains the identity attributes needed for the user request.
5. The service provider consumes the assertion or token to establish a security context for the user.
6. The service provider validates the assertion and authorizes the resource request.
7. The service provider establishes a session with the user.

CloudAccess can also provide authentication when users initiate access to the application from the service provider.

The following describes the SSO experience for trusted access to an application initiated from the service provider:

1. The user attempts to log in to an application.
2. The login is redirected to CloudAccess.
3. CloudAccess prompts the user for the user name and password. Or, if Kerberos is configured, CloudAccess performs seamless authentication.
4. CloudAccess verifies the user name and password using the identity sources. Or, if Kerberos is configured, CloudAccess validates the Kerberos token.
5. CloudAccess provides an assertion to the application service provider.
6. The service provider validates the assertion and allows the user to access the application.

1.1.2 Basic Single Sign-On

Basic single sign-on provides an internal credentials store where users can save their credentials for third-party websites that require a password to be sent at login. The destination website’s login page must use HTML Forms as the main point of interaction with the user. A user typically has a site-specific user name and password for each destination website. CloudAccess stores the user’s credentials for each site in AES-256 encrypted format. After a user authenticates to CloudAccess, the user can access a website without manually re-entering credentials for the site.

CloudAccess provides many connectors for Basic Single Sign-on (SSO). For more information, see Section 1.4, “Connectors for Basic Single Sign-On,” on page 14.

1.1.3 OAuth 2.0 Single Sign-On

OAuth 2.0 single sign-on provides simple authenticated access to a protected web service through CloudAccess. CloudAccess behaves as an OAuth 2.0 Authorization Server and Resource Server to provide user authentication and all OAuth2 token creation and validation for access. It uses the Authorization Code flow as detailed in the *OAuth 2.0 Authorization Framework (IETF RFC 6749)* (http://tools.ietf.org/html/rfc6749#section-4.1) document.

CloudAccess supports OAuth 2.0 access in service-provider mode. End users can access the protected resource by browsing to the URL of the OAuth client application. For example, users can enter the URL directly into the browser and be redirected to log in to CloudAccess, or they can use a bookmark or the landing page appmark after logging in to CloudAccess.

The following describes the experience for OAuth 2.0 access to an application by browsing to the URL:

1. The user accesses the protected resource by entering the URL directly in the browser.
2. The user is redirected to the CloudAccess login page.
3. The user provides login credentials to CloudAccess, such as his corporate user name and password.
4. CloudAccess authenticates the user’s credentials against the identity sources.
5. CloudAccess validates the OAuth2 token for the client.
6. The user gains access to the resource.

The following describes the experience for OAuth 2.0 access to an application through CloudAccess:

1. Users provide login credentials directly to CloudAccess, such as their corporate user name and password.
2. CloudAccess authenticates the user’s credentials against the identity sources.
3. CloudAccess presents the landing page to the user with links to applications that the user is entitled to use.
4. The user clicks the bookmark or the landing page appmark for the application.
5. CloudAccess validates the OAuth2 token for the client.
6. The user gains access to the resource.

1.1.4 Bookmarks

In CloudAccess, you can create bookmarks to access web applications through CloudAccess that do not require additional passwords. The bookmarks are accessible from the browser landing page or directly from the MobileAccess app on users’ mobile devices.

The following describes the experience for bookmark access to a web application through CloudAccess:

1. Users provide login credentials directly to CloudAccess, such as their corporate user name and password.
2. CloudAccess authenticates the user’s credentials against the identity sources.
3. CloudAccess presents the landing page to the user with links to applications that the user is entitled to use.
4. The user clicks the appmark for the bookmark.
5. The user gains access to the resource.

1.2 Connectors for Federated Single Sign-On and Provisioning

CloudAccess provides several connectors that enable federated single sign-on and logout as well as account provisioning. The connectors ship with the appliance. You can use these connectors if you have a CloudAccess license as well as an account with the destination service.

- Chapter 9, “Connector for Google Apps,” on page 63
- Chapter 10, “Connector for Microsoft Office 365,” on page 69

- Chapter 11, “Connector for Salesforce,” on page 79
- Chapter 12, “Connector for ServiceNow,” on page 89

After you initialize the appliance, the connectors for Google Apps, Salesforce, and ServiceNow are automatically visible in the **Applications** palette in the administration console. However, the connector for Office 365 is not visible in the palette until you install the connector on the Windows Management Server.

Provisioning is available for users in your corporate identity sources for Active Directory, eDirectory, and JDBC. You must map authorizations for the appropriate roles (groups) to enable their entitlements to the applications. Users must log in with a corporate identity to access their provisioned account.

1.3 Connectors for Federated Single Sign-On

CloudAccess provides additional connectors that you can use for federated single sign-on to web services and applications through CloudAccess using the SAML 2.0 protocol. You can download the additional connectors from the Application Connector Catalog through CloudAccess. For configuration information, see Chapter 13, “Single Sign-On Connectors,” on page 93.

You can also create custom connectors for federated single sign-on and logout using the Access Connector Toolkit. For more information, see Chapter 3, “Creating Custom Connectors,” on page 27.

After you download a connector or create a custom connector, you must import it to CloudAccess to make it available in the **Applications** palette in the CloudAccess administration console. You can use these connectors if you have a CloudAccess license as well as an account with the destination service.

1.4 Connectors for Basic Single Sign-On

CloudAccess provides many connectors for Basic Single Sign-on (Basic SSO). They allow users to access web services that use forms-based authentication and require that the user’s password be sent at login. Examples include social media sites such as Evernote, LinkedIn, and Facebook. Basic SSO connectors work with the Basic SSO extension for supported browsers running on the user’s computer.

CloudAccess supports using multiple connectors for Basic SSO. Each instance points to a different destination website. You can use these connectors if you have a CloudAccess license. Users have individual accounts with the destination services.

Connectors for Basic SSO are available in the Application Connector Catalog. You can browse or search the catalog for appropriate connectors to import to your appliance. You can access the catalog from the **Applications** palette in the administration console.

You can also create custom connectors for Basic SSO using the Access Connector Toolkit. For more information, see Chapter 3, “Creating Custom Connectors,” on page 27. After you create a custom connector, you must import it to CloudAccess to make it available in the **Applications** palette in the CloudAccess administration console. For more information, see Chapter 8, “Connectors for Basic SSO,” on page 57.

1.5 Connector for OAuth 2.0 Single Sign-On

CloudAccess provides a connector for OAuth2 Resources that allows single sign-on with simple OAuth 2.0 authenticated access to a protected web service through CloudAccess. The connector ships with the appliance.

CloudAccess supports using multiple instances of the connector for OAuth2 Resources. Each instance points to a different destination OAuth 2.0 resource, or to a set of OAuth 2.0 resources that have the same authentication requirements. You can use this connector if you have a CloudAccess license as well as an account with the destination service.

For more information, see Chapter 7, “Connector for OAuth2 Resources,” on page 53.

1.6 Connector for Bookmarks

The connector for Bookmarks is a container for simple bookmarks to applications that do not require additional passwords for access. The connector ships with the appliance. You can use this connector if you have a CloudAccess license as well as access to the destination web service.

For more information, see Chapter 6, “Connector for Bookmarks,” on page 51.

1.7 Custom Connectors

CloudAccess provides the Access Connector Toolkit (ACT) that allows you to create custom connectors. If you need help creating a custom connector to use with CloudAccess, Priority Support customers have the option to open a service request with Technical Support (http://www.netiq.com/support). NTS is available to provide toolkit support as well as to configure the connectors to work with integrated applications. Additional information from the SaaS provider is usually required.

NOTE: Before you contact Technical Support, please complete the appropriate worksheet for the connector type that you want to create. See Appendix A, “Custom Connector Worksheets,” on page 109.

The Access Connector Toolkit facilitates custom connector development efforts without coding or scripting. You can create connectors for identity-aware SaaS applications that support federated single sign-on and logout or that support basic single sign-on. You can use the toolkit and custom connectors if you have a CloudAccess license as well as appropriate accounts with the destination services.

For more information, see Chapter 3, “Creating Custom Connectors,” on page 27.

2 Configuring Connectors

CloudAccess provides many connectors to web services and applications. This section describes configuration tasks that are common to multiple connectors.

- Section 2.1, "Requirements for Connectors," on page 17
- Section 2.2, "Viewing Connectors for Applications," on page 17
- Section 2.3, "Providing Access to Applications for Users," on page 18
- Section 2.4, "How CloudAccess Provisions User Accounts," on page 18

2.1 Requirements for Connectors

As you configure connectors, ensure that you meet these general setup requirements:

- ❒ The **Display Name** for each configured instance of a connector must be unique for the appliance. The name allows you to identify a configured instance of the connector on the Admin page.
- ❒ The **Federation Instructions** on a connector's Configuration page provide the information that you will use to configure federation for CloudAccess on the service provider site. The information identifies where on the service provider's site to find the federation configuration capability as well as the field values and other guidance that you need to complete the required information.

  When you configure the connector, the federation instructions automatically provide the following information about your appliance as the identity provider:

  - The URL for single sign-on

    `https://appliance_dns_name/osp/a/t1/auth/saml2/sso`
  - The URL for single logout

    `https://appliance_dns_name/osp/a/t1/auth/app/logout`
  - The URL for the identity provider's entityID

    `https://appliance_dns_name/osp/a/t1/auth/saml2/metadata`
  - The X.509 signing certificate for the appliance

    The web service or application uses the certificate to set the trust relationship with CloudAccess.

  NOTE: When you copy the appliance's signing certificate, ensure that you include all leading and trailing hyphens in the certificate's Begin and End tags.

2.2 Viewing Connectors for Applications

CloudAccess displays the connectors on the Admin page of the administration console:

- **Applications palette:** Displays unconfigured connectors that ship with the appliance or that you have downloaded or imported.
- **Applications panel:** Displays configured connectors for the web services or applications that you want to make available to users.

2.3 Providing Access to Applications for Users

CloudAccess provides a portal page for users that consists of a login page and a landing page. The portal page exposes many of the features of the appliance for users, such as SSUS and Google reCAPTCHA.

The URL for the portal page is the public DNS name of the CloudAccess appliance:

```
https://CloudAccess_appliance_dns_name.com
```

Provide this URL to your users to access the resources provided through the CloudAccess appliance.

If you have configured the One-Time Password (OTP), users will see a different page first. This page is the authentication code page where users enter their one-time password. For more information, see "Configuring the TOTP Tool for Two-Factor Authentication Using Google Authenticator" in the *CloudAccess Installation and Configuration Guide*.

2.4 How CloudAccess Provisions User Accounts

The connector for Google Apps, the connector for Salesforce, the connector for ServiceNow, and the connector for Microsoft Office 365 support both account provisioning and single sign-on. These connectors are delivered with the appliance. You can use these connectors if you have a CloudAccess license as well as an account with the destination service.

- Section 2.4.1, "Requirements for Provisioning," on page 18
- Section 2.4.2, "Understanding Provisioning," on page 19
- Section 2.4.3, "Samples of Account Creations," on page 20
- Section 2.4.4, "CloudAccess Default Naming Convention for Newly Provisioned Accounts," on page 21
- Section 2.4.5, "Matching Criteria for Merging Existing Accounts," on page 21
- Section 2.4.6, "Understanding Naming Policy Options," on page 24
- Section 2.4.7, "Understanding Collision Handling in Provisioning," on page 25

2.4.1 Requirements for Provisioning

The connectors that support SSO and provisioning can provision accounts for users in your corporate identity sources for Active Directory, eDirectory, and JDBC. For CloudAccess to provision user accounts to the SaaS applications, each user account in the identity source must contain the attributes listed. If you are using an LDAP identity source, there are specific attributes that must be populated on the user accounts. If you are using a JDBC database, there are certain columns of information that must be populated. The information for each identity source is different. For more information about identity source requirements for provisioning, see "Configuring LDAP and JDBC Identity Sources" in the *CloudAccess Installation and Configuration Guide*.

By default, the connectors use the user name when provisioning users to the SaaS applications. However, you can change the naming policy to match accounts that might already exist in the SaaS applications, and let CloudAccess use them for management and providing SSO. For more information, see Section 2.4.6, "Understanding Naming Policy Options," on page 24.

IMPORTANT: We recommend reviewing and, if appropriate for your environment, changing the default naming policy *before* you provision users to the SaaS applications. If you change a naming policy after users have already been provisioned, you must either remove the users from the policy mapped group for the SaaS account, and then put the users back in the group, or undo the policy mapping and then redo it. Depending on the order of operations, it might take a couple of tries for the accounts to be completely changed in the SaaS application.

You must map authorizations for the appropriate roles (groups) to enable their entitlements to the applications. For more information, see “Mapping Authorizations” in the *CloudAccess Installation and Configuration Guide*. Users must log in with a corporate identity to access their provisioned account.

IMPORTANT: The connectors cannot provision accounts for users in a Self-Service User Store (SSUS) identity source, a SAML 2.0 Inbound (SAML2 In) internal identity store, or social identity sources such as Facebook. You should not create policy mappings between provisioning applications and SSUS, SAML2 In, or social (Facebook, etc.) identity sources.

2.4.2 Understanding Provisioning

CloudAccess creates a new account or merges an existing account in the SaaS applications for users who are members of groups in the identity sources that you map for entitlements to the applications. This is called *provisioning*.

Account provisioning occurs in one of the following ways:

- **Automatic:** CloudAccess provisions an account for users who are members of groups that are entitled to use the application.
  - If approval is not required when you configure policy mapping, an account is provisioned when you map authorizations for the SaaS applications to the identity source roles.
  - If approval is required, the account is not provisioned until the request is approved.
- **User-controlled (Salesforce only):** You can configure the connector for Salesforce to allow users control of when their accounts are provisioned. A configuration option is available on the connector to **Prompt users for an existing account before provisioning**.

  When you select this option, users have two choices during their initial attempt to access the Salesforce application using single sign-on through CloudAccess:
  - **I do not have an existing account. Create one for me.**
  - **I already have an existing account. These are my credentials:**

When it provisions a new account, CloudAccess determines if a matching user account already exists in the SaaS application. This search occurs whether the provisioning method is automatic or user controlled.

- If a match is found, CloudAccess merges the user with the existing SaaS application account. For more information, see Section 2.4.5, “Matching Criteria for Merging Existing Accounts,” on page 21.
- If no match is found, CloudAccess creates a new account. For more information, see Section 2.4.4, “CloudAccess Default Naming Convention for Newly Provisioned Accounts,” on page 21.

Whether CloudAccess merges the user account or creates a new account, the user’s SaaS application password is set to a random value that CloudAccess generates. The user’s authentication to the SaaS application now uses federated single sign-on with SAML 2.0. CloudAccess sends information about an authenticated user to the destination application using an assertion or token. The user does not log in directly to the application. For information about single sign-on for users, see Section 2.3, “Providing Access to Applications for Users,” on page 18.

2.4.3 Samples of Account Creations

The examples in this section describe the experience for automatic account creation and user-controlled provisioning (that is, when the **Prompt users for an existing account before provisioning** option is enabled on the connector for Salesforce).

Sample experience with the automatic account provisioning:

1. The administrator maps one or more SaaS authorizations to the identity source role (group).
2. (Conditional) If approval is required, the person with the Approval role approves or denies the account creation.
3. CloudAccess searches for an existing, matching account.
4. CloudAccess merges an existing matching account. If CloudAccess finds a matching account, CloudAccess merges the existing account with the new account it creates for the user. For example, the user's mail attribute in the identity source matches a user name in the Salesforce domain.

   Or

   CloudAccess provisions a new account. If CloudAccess does not find a matching account, CloudAccess creates a new account for the user in the SaaS application, following the naming conventions in Table 2-1 on page 21.
5. Users log in to CloudAccess with their corporate credentials, and CloudAccess authenticates them against the identity sources.
6. CloudAccess presents users with appmarks for the SaaS applications they are entitled to access.
7. Users click the appmark for the entitled SaaS application they want to use.
8. CloudAccess uses single sign-on to authenticate the users to the SaaS application.

Sample experience with user-controlled account provisioning (Salesforce only):

1. The administrator selects the **Prompt users for an existing account before provisioning** option when configuring the connector for Salesforce.
2. The administrator maps one or more Salesforce authorizations to the identity source role (group) and grants approval, if required.
3. (Conditional) The person with the Approval role approves or denies the account creation.
4. Users log in to CloudAccess with their corporate credentials, and CloudAccess authenticates them against the identity sources.
5. CloudAccess presents users with appmarks for the Salesforce applications they are entitled to access.
6. Users click the appmark for the entitled application they want to use.
7. CloudAccess presents two options to users:
   - **I do not have an existing account. Create one for me.**
   - **I already have an existing account. These are my credentials:**
8. Regardless of the option the user selects, CloudAccess searches for an existing, matching account.
9. CloudAccess merges an existing matching account. If CloudAccess finds a matching account, CloudAccess merges the existing account with the new account it creates for the user. For example, the user's mail attribute in the identity source matches a user name in the Salesforce domain.

Or

CloudAccess provisions a new account. If CloudAccess does not find a matching account, CloudAccess creates a new account for the user in Salesforce, following the naming conventions in Table 2-1 on page 21.

10. CloudAccess uses single sign-on to authenticate the users to Salesforce.

2.4.4 CloudAccess Default Naming Convention for Newly Provisioned Accounts

CloudAccess contains a default naming convention for creating the user accounts in the SaaS applications.

Table 2-1 *CloudAccess Naming Convention*

Identity Source	Connector for Google Apps	Connector for Microsoft Office 365	Connector for Salesforce	Connector for ServiceNow
Active Directory	sAMAccountName	sAMAccountName@*Federated Domain Name*	sAMAccountName@*Salesforce Domain Name*	CN
eDirectory	CN	CN@*Federated Domain Name*	CN@*Salesforce Domain Name*	CN
JDBC	CN	CN@*Federated Domain Name*	CN@*Salesforce Domain Name*	CN

As shown in Table 2-1, the user name attribute for Office 365 and Salesforce are in the form of an email address. For Office 365, CloudAccess creates a unique user name based on the sAMAccountName or CN of the user object in the identity source prepended to the federated domain name configured for the organization in the Office 365 account. For Salesforce, CloudAccess creates a unique user name based on the sAMAccountName or CN of the user object in the identity source prepended to the domain name configured in the company profile at the Salesforce account.

2.4.5 Matching Criteria for Merging Existing Accounts

The following sections define the matching criteria of the connectors that provision users:

- "Matching Criteria for the Connector for Google Apps" on page 21
- "Matching Criteria for the Connector for Office 365" on page 22
- "Matching Criteria for the Connector for Salesforce" on page 22
- "Matching Criteria for the Connector for ServiceNow" on page 23

Matching Criteria for the Connector for Google Apps

Table 2-2 contains the matching criteria for the connector for Google Apps. CloudAccess compares the Google Apps email attribute value to the listed identity source attribute value. If there is a match, CloudAccess merges the accounts. If there is no match, CloudAccess creates a new account in Google Apps. The display name is the user-friendly name for the attribute parameter that it represents.

Table 2-2 *Google Apps Matching Criteria*

Source	Display Name	Attribute Name
Google Apps	Email	Username
Active Directory	User logon name	sAMAccountName
eDirectory	Username	CN
JDBC	Username	CN

Matching Criteria for the Connector for Office 365

Table 2-3 contains the matching criteria for the connector for Office 365. CloudAccess compares the Office 365 `userPrincipalName` attribute value with the value in the identity source attribute plus the @ sign plus the Office 365 federated domain name. If the values match, CloudAccess merges the accounts. If there is no match, CloudAccess creates a new account in Office 365. The display name is the user-friendly name for the attribute parameter that it represents.

Table 2-3 *Office 365 Matching Criteria*

Source	Display Name	Attribute Name
Office 365	User name	userPrincipalName (upn)
Active Directory	User logon name@*Federated Domain Name*	sAMAccountName@*Federated Domain Name*
eDirectory	Username@*Federated Domain Name*	CN@*Federated Domain Name*
JDBC	Username@*Federated Domain Name*	CN@*Federated Domain Name*

Matching Criteria for the Connector for Salesforce

Table 2-4 contains the matching criteria for the connector for Salesforce. CloudAccess matches on three different attributes in a priority order. If CloudAccess does not find a match for the value in the first attribute, it performs a search on the value of the second attribute, and if it does not find a match, it performs the search for the value in the third attribute. The display name is the user-friendly name for the attribute parameter that it represents.

Table 2-4 *Salesforce Matching Criteria*

Source	Display Name	Attribute Name
First Priority		
Salesforce	Federation identifier	N/A
Active Directory	N/A	objectGUID
		employeeID
eDirectory	N/A	GUID
		workforceID

Source	Display Name	Attribute Name
JDBC	N/A	N/A
Second Priority		
Salesforce	Username	Username
Active Directory	User logon name@*Salesforce Domain Name*	sAMAccountName@*Salesforce Domain Name*
eDirectory	Username@*Salesforce Domain Name*	CN@*Salesforce Domain Name*
JDBC	Username@*Salesforce Domain Name*	CN@*Salesforce Domain Name*
Third Priority		
Salesforce	Username	Username
Active Directory	E-mail	Mail
eDirectory	E-mail Address	Internet EMail Address
JDBC	N/A	N/A

Matching Criteria for the Connector for ServiceNow

Table 2-5 contains the matching criteria for the connector for ServiceNow. CloudAccess matches on two different attributes in a priority order. CloudAccess tries to match existing accounts first by CN, then by email address. If CloudAccess does not find a match for the value in the first attribute, it performs a search on the value of the second attribute. The display name is the user-friendly name for the attribute parameter that it represents.

Table 2-5 *ServiceNow Matching Criteria*

Source	Display Name	Attribute Name
First Priority		
ServiceNow	Username	CN
Active Directory	User logon name	sAMAccountName
eDirectory	Username	CN
JDBC	Username	CN
Second Priority		
ServiceNow	Email	Email
Active Directory	Email	Mail
eDirectory	Email-address	Internet Email address
JDBC	Email	Email

2.4.6 Understanding Naming Policy Options

Before you map users to the appropriate applications to set their entitlements, you can specify the naming policy that the SaaS connectors should use for each user account when provisioning users.

Naming policies are used for both newly provisioned accounts and matching accounts. New accounts are named according to the naming policy you specify. For an existing account, the connector tries to match it, and if it finds a matching account, it renames the account if necessary to match the policy.

NOTE: CloudAccess does not currently have any selection criteria for the naming policy for ServiceNow. Whatever the CN attribute is when the user is imported, that is the user name that is created in ServiceNow.

The user name attribute for Office 365 is in the form of an email address. If you do not specify a naming policy, by default the connector creates a unique user name based on the sAMAccountName or CN of the user object in the identity source prepended to the federated domain name configured for the organization in the Office 365 account.

The following table shows how the connector for Office 365 handles an example user named John Q Public, with the samAccountName jpublic in Active Directory, depending on the naming rule you select.

Table 2-6 *Rule Options*

Option	AD Attributes	Office 365 Account (in the ag4c.com domain)
UserName (default option)	samAccountName	jpublic@ag4c.com
Firstname.Lastname	firstname + lastname	john.public@ag4c.com
FirstInitial + Lastname	substring(1)firstname + lastname	jpublic@ag4c.com
Firstname + MiddleInitial + Lastname		johnqpublic@ag4c.com

The SaaS connectors can usually handle certain special characters, such as apostrophes (‘), double quotes (“), and back-ticks (‘) in user names. The connectors handle spaces and apostrophes as follows:

- If any of the values, such as the middle initial, are blank in the identity source, the connectors ignore them and leave them out of the name sent to the SaaS applications.
- **Office 365:** If you have a user name containing an apostrophe, the connector will change the user account and keep the apostrophe in place. For example, if you set the naming rule to firstname.lastname and you have a user name of Mike O’Shea, the connector will change the user account to Mike.O’Shea@ag4c.com. The email address will also include the apostrophe.
- **Google Apps:** If you have a user name containing an apostrophe, the connector will strip out that character. For example, if you set the naming rule to firstname.lastname and you have a user name of Mike O’Shea, the connector will send the account name oshea to Google.
- **Salesforce:** The connector for Salesforce is currently unable to handle apostrophes in the user name. We recommend that you remove any apostrophes in username, firstname, and lastname to avoid provisioning failures.
- **ServiceNow:** The connector for ServiceNow handles spaces, apostrophes, and other special characters as expected. For example, the connector creates john smith as john smith, and creates karen.o’malley as karen.o’malley.

If you change a user name in the identity source, such as jpublic to john.public, the Office 365 user will not lose any existing email, and the user's new email address will be john.public@ag4c.com. However, it might take a few minutes for the mailbox to be ready to accept emails after a name change.

2.4.7 Understanding Collision Handling in Provisioning

CloudAccess handles collisions during provisioning using the **Veto** method. If there is a duplicate user name, the first user is provisioned, but the second user is not. For example, if you have a user named John Q. Public and a user names Jack D. Public, and your naming policy is set to FirstInitial + Lastname, the workflow is as follows for Google Apps (assuming John is provisioned first):

1. CloudAccess provisions jpublic@ag4c.com as expected.
2. The connector searches for jpublic and finds a match. However, since he has already been provisioned, the connector recognizes that this is not the same account and shows an error in the googleapps connector logs similar to the following:

```
>      DirXML Log Event -------------------
>      Driver:   \IDVAULT\system\driverset1\connectors_GOOGLEAPPS_VXE3T
>      Channel:  Subscriber
>      Object:   \IDVAULT\data\users\jackpublic
>      Status:   Error
>      Message:  Code(-9063) Object matching policy found an object that
> is already associated: data\users\johnpublic
```

   where `johnpublic` is the existing object that was provisioned first.

You can download the logs for the SaaS connectors using the troubleshooting tools in CloudAccess. For more information, see Section 14.1, “Using Troubleshooting Tools for Application Access Issues,” on page 99.

NOTE: The potential for collisions increases with some of the naming policy options, such as firstname.lastname. Consider carefully whether you need to change the default naming policy of user name. Since the CloudAccess namespace is a flat tree (all users go into the ou=users container) and all names must be unique, the potential for collisions when using the default naming option is eliminated.

The connector can handle up to 999 collisions. After 999 collisions, the connector stops provisioning new accounts and logs an error in the connector log file, with a message similar to the following: “A user with this user principal name already exists.”

3 Creating Custom Connectors

CloudAccess provides the Access Connector Toolkit (ACT) that allows you to create custom connectors. If you need help creating a custom connector to use with CloudAccess, Priority Support customers have the option to open a service request with Technical Support (http://www.netiq.com/support). Technical Support is available to provide toolkit support as well as to configure the connectors to work with integrated applications. Additional information from the SaaS provider is usually required.

NOTE: Before you contact Technical Support, please complete the appropriate worksheet for the connector type that you want to create. See "Custom Connector Worksheets."

The Access Connector Toolkit facilitates custom connector development efforts without coding or scripting. You can create custom connectors for identity-aware web service or applications that use the following authentication methods for single sign-on:

- SAML 2.0
- WS-Federation
- SAML 2.0 Inbound (SAML2 In)
- Basic SSO (forms-based)

After you create a connector, you must export it from the toolkit as a file that you can import into CloudAccess. You can use the CloudAccess administration console to import and enable the connector, and to create appmarks and to map policies for the web service or application.

- Section 3.1, "Accessing the Access Connector Toolkit," on page 27
- Section 3.2, "Toolkit Requirements," on page 28
- Section 3.3, "Creating a SAML 2.0 Connector Template," on page 28
- Section 3.4, "Creating a WS-Federation Connector Template," on page 30
- Section 3.5, "Creating a SAML 2.0 Inbound (SAML2 In) Connector Template," on page 31
- Section 3.6, "Creating a Basic SSO Connector Template," on page 35
- Section 3.7, "Modifying a Connector," on page 37
- Section 3.8, "Exporting a Connector Template," on page 38
- Section 3.9, "Importing and Configuring Custom Connectors," on page 38

3.1 Accessing the Access Connector Toolkit

The Access Connector Toolkit is a web application that you access through the CloudAccess appliance. Log in to the toolkit using CloudAccess administrator credentials at:

```
https://appliance_dns_name/css/toolkit
```

The Access Connector Toolkit does not currently provide a logout option, though the session does time out after 60 minutes of inactivity. Ensure that you close the browser after you finish working in the Access Connector Toolkit.

3.2 Toolkit Requirements

The Access Connector Toolkit is a web application that ships with CloudAccess. You can use the Access Connector Toolkit to create custom connectors if you have a CloudAccess license as well as appropriate accounts with the destination services.

3.2.1 Toolkit Compatibility

The Access Connector Toolkit contains new functionality in CloudAccess 2.1 and later releases. To update an existing custom connector template with the new functions, you can import the template into the new toolkit, and then export the template again. The updated connector template contains the new functionality.

Templates that you create with the new toolkit are not backwards compatible with prior releases of the toolkit. You cannot import a connector from CloudAccess 2.1 or later into a toolkit that came with a prior version of CloudAccess. The import fails.

3.2.2 Provisioning Support

Provisioning is supported only through connectors created by NetIQ. At this time, you cannot create a custom connector template that supports provisioning user accounts to the connected system.

Account provisioning is not supported for users in a SAML 2.0 Inbound (SAML2 In) unmanaged internal identity store or in a Self-Service User Store (SSUS) identity source. For more information, see Section 2.4.1, “Requirements for Provisioning,” on page 18.

3.3 Creating a SAML 2.0 Connector Template

To create a connector for single sign-on with SAML 2.0, you can use the **SAML2** option in the Access Connector Toolkit.

- Section 3.3.1, “SAML 2.0 Requirements for the Application Service Provider,” on page 28
- Section 3.3.2, “Planning for a SAML 2.0 Connector,” on page 29
- Section 3.3.3, “Creating a SAML 2.0 Connector Template for an Application,” on page 29

3.3.1 SAML 2.0 Requirements for the Application Service Provider

To create a custom SAML 2.0 connector for a destination application, ensure that the service provider meets the following protocol-specific requirements:

❒ Supports identity federation using the SAML 2.0 protocol.

For more information about SAML, see the OASIS website (https://wiki.oasis-open.org/security/FrontPage).

❒ Supports the SAML web browser single sign-on profile, with the Redirect and POST bindings for service-provider-initiated SSO, and the POST binding for identity-provider-initiated SSO.

❒ Provides a capability in the application's administration console that allows you to enable and configure SAML SSO with CloudAccess as the identity provider.

❒ Provides technical documents that describe the application's SAML federation requirements, metadata, and assertions.

3.3.2 Planning for a SAML 2.0 Connector

Before you attempt to create the SAML 2.0 connector, you must collect information about the destination web service or application. Ask the application service provider the following types of questions to gather the required information:

- What does your SAML assertion look like?
- Do you have a SAML metadata document? What fields, if any, are customer-specific?
- Does your service support the SAML single logout protocol?
- What are the required configuration steps in your application to set up federation?
- What information do you provide to customers when they are setting up federation with their identity source?

NOTE: You can use a worksheet to organize the information. See "Worksheet for SAML or WS-Federation Custom Connectors".

3.3.3 Creating a SAML 2.0 Connector Template for an Application

A SAML 2.0 connector template consists of multiple components for federation, metadata, and assertion information.

To create a custom SAML 2.0 connector:

1. Log in as an administrator to the Access Connector Toolkit.
2. Click **New** > **SAML2**.

   The connector **Type** is `SAML2`. The **Type Name** is `Generic SAML2 Connector`.
3. On the **Template** tab, provide the following information:
   - Template properties
   - Whether the service provider requires a signing certificate
   - Federation instructions for the service provider
   - New settings that need to be collected on the Configuration page of the connector
4. Click the **Metadata** tab, then use one of the following methods to specify the metadata:
   - Select **Request**, then specify the source URL to retrieve the metadata.
   - Complete the fields to manually generate the metadata.
   - Import the values from a file or URL, and modify them for your deployment environment.
5. Click the **Assertion** tab, then define the properties and attributes required for the assertion.

   5a On the **Properties** subtab, specify the properties for the assertion.

   5b On the **Attributes** subtab, click **New**, specify and define the identity attribute, then click **Save**.

   5c (Conditional) If the service provider requires other identity attributes for an assertion, repeat Step 5b to map the SAML assertion attribute to an attribute in your identity source.

6 (Optional) If it is supported, create the provisioning definitions. For more information, see Section 3.2.2, “Provisioning Support,” on page 28.

7 Click **Save** to save the new connector template.

8 Proceed to Section 3.8, “Exporting a Connector Template,” on page 38 to finish creating the new connector.

3.4 Creating a WS-Federation Connector Template

To create a connector for single sign-on with WS-Federation, you can use the **WS-Fed** option in the Access Connector Toolkit.

- Section 3.4.1, “WS-Federation Requirements for the Application Service Provider,” on page 30
- Section 3.4.2, “Planning for a WS-Federation Connector,” on page 30
- Section 3.4.3, “Creating a WS-Federation Connector Template for an Application,” on page 31

3.4.1 WS-Federation Requirements for the Application Service Provider

To create a custom WS-Federation connector for a destination application, ensure that the service provider meets the following protocol-specific requirements:

❒ Supports identity federation using the WS-Federation protocol.

For more information about WS-Federation, see the OASIS website (http://docs.oasis-open.org/wsfed/federation/v1.2/os/ws-federation-1.2-spec-os.html) or see the MSDN Library article (http://msdn.microsoft.com/en-us/library/bb498017.aspx).

❒ Supports the WS-Federation Passive Requestor Profile.

❒ Provides a capability in the application’s administration console that allows you to enable and configure WS-Federation SSO.

❒ Provides technical documents that describe the application’s WS-Federation federation requirements, metadata, and security tokens.

3.4.2 Planning for a WS-Federation Connector

Before you attempt to create a WS-Federation connector, you must collect information about the destination web service or application. Ask the web service or application vendors the following types of questions to gather the require information:

- What does your WS-Federation security token look like?
- Do you have a WS-Federation metadata document? What fields, if any, are customer-specific?
- What are the required configuration steps in your application to set up federation?
- What is the information that you provide to customers when they are setting up federation with their identity source?

NOTE: You can use a worksheet to organize the information. See “Worksheet for SAML or WS-Federation Custom Connectors”.

3.4.3 Creating a WS-Federation Connector Template for an Application

A WS-Federation connector template consists of multiple components for federation, metadata, and assertion information.

To create a custom connector:

1. Log in as an administrator to the Access Connector Toolkit.
2. Click **New** > **WSFed**.

   The connector **Type** is `WS-Fed`. The **Type Name** is `Generic WS-Fed Connector`.
3. On the **Template** tab, provide the following information:
   - Template properties
   - Whether the service provider requires a signing certificate
   - Federation instructions for the service provider
   - New settings that need to be collected on the Configuration page of the connector
4. Click the **Metadata** tab, then use one of the following methods to specify the metadata:
   - Select **Request**, then specify the source URL to retrieve the metadata.
   - Complete the fields to manually generate the metadata.
   - Import the values from a file or URL, and modify them for your deployment environment.
5. Click the **Assertion** tab, then define the properties and attributes required for the security token.

   5a On the **Properties** subtab, specify the properties for the assertion.

   5b On the **Attributes** subtab, click **Predefined**, click the identity attribute, modify the definition if needed, then click **Save**.

   If a predefined option does not exist, use **New** to define it.

   5c (Conditional) If the service provider requires other identity attributes for an assertion, repeat Step 5b to map the WS-Federation attribute to an attribute in your identity source.
6. (Optional) Create the provisioning definitions. For more information, see Section 3.2.2, “Provisioning Support,” on page 28.
7. Click **Save** to save the new connector template.
8. Proceed to Section 3.8, “Exporting a Connector Template,” on page 38 to finish creating the new connector.

3.5 Creating a SAML 2.0 Inbound (SAML2 In) Connector Template

To allow the appliance to be a SAML 2.0 service provider, you can create a SAML 2.0 Inbound connector using the Access Connector Toolkit. SAML2 Inbound as an identity source is not available during initialization of the appliance. However, after you export the connector and import it in the appliance, the SAML2 In connector appears as an identity source. You configure an instance of the identity source with information about an appropriate identity provider to enable the service provider functionality of the appliance, and to allow the identity provider to send a SAML token to the appliance using the SAML 2.0 POST profile.

After you configure the SAML2 In identity source, the appliance login page provides a link to the login page of the SAML 2.0 identity provider, located to the left of the user name and password login options. The SAML 2.0 users log in through the identity provider to gain access to the appliance landing page.

- Section 3.5.1, "Understanding SAML2 In Identity Sources," on page 32
- Section 3.5.2, "Requirements for Using SAML2 In Identity Sources," on page 33
- Section 3.5.3, "SAML2 In Requirements for the Identity Provider," on page 34
- Section 3.5.4, "Planning for a SAML2 In Connector," on page 34
- Section 3.5.5, "Creating a SAML2 In Connector for an Identity Provider," on page 35

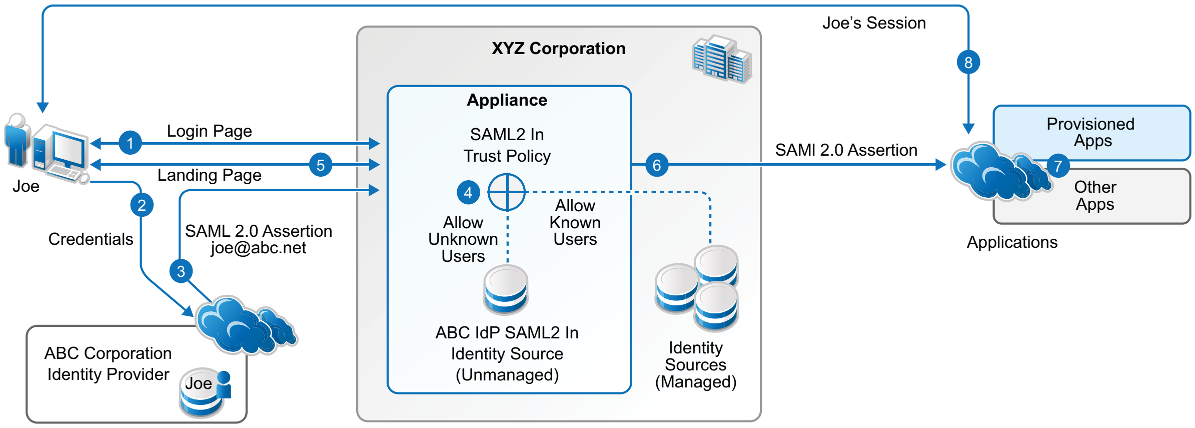
3.5.1 Understanding SAML2 In Identity Sources

A SAML2 In identity source can trust assertions from users who log in through the SAML2 In identity provider as known users or unknown users.

When you configure the SAML2 In identity source to accept assertions only for *known users*, the identity source accepts the authentication for users from the SAML2 In identity provider who have a matching email address in any one of the managed identity sources (for example, eDirectory, Active Directory, or JDBC). Matching will not occur for users created with SSUS. The identity source denies access to the unmatched users. The known users can access applications according to the authorizations you map for their roles (groups) in the managed identity sources. Account provisioning and application access works the same as when the user logs in with corporate credentials through the appliance.

When you configure the SAML2 In identity source to accept assertions only for *unknown users*, the identity source accepts the authentication for all users from the SAML2 In identity provider, which should be a unique set of users as defined by email addresses. The identity source creates unique user objects in an unmanaged internal identity store for the identity provider. The unknown users can access applications according to the authorizations you map for their roles (groups) in the SAML2 In identity source.

Figure 3-1 *Using the Appliance as a Service Provider with the SAML2 In Identity Source*



The following is the experience for the SAML2 In user:

1. The user goes to the appliance login page, then clicks the link for the identity provider.
2. The user receives the login page from the identity provider, and sends credentials.
3. The identity provider authenticates the user, then sends a SAML 2.0 assertion via the user's browser, to the appliance using the SAML 2.0 POST profile. The user identity is based on an email address.
4. The SAML2 In identity source applies the trust policy that you configured for the identity provider:
    a. **Allow access for unknown users:** The appliance accepts authentication for all users from the identity provider, and creates a unique user object for the user in an internal identity store, based on the email address in the assertion.

    b. **Allow access for known users:** The appliance accepts authentication for users who have a matching identity in any of the managed identity sources, based on the email address in the assertion.
5. The appliance sends the landing page to the authenticated user.
6. When the user selects an appmark, the appliance builds an assertion for the user identity based on the SAML2 In trust policy:
    - **Allow access for unknown users:** The user's identity attributes are based on the user's information in the SAML2 In unmanaged internal identity store for the identity provider.
    - **Allow access for known users:** The user's identity attributes are based on the user's information in one of the managed identity sources.
7. The user accesses applications based on the entitlements that are associated with his logged-in identity.
8. The application service provider establishes a session with the user.

3.5.2 Requirements for Using SAML2 In Identity Sources

Consider the following requirements when you configure a SAML2 In identity source to allow access only for unknown users:

- ❒ The users who authenticate through the SAML2 In identity provider have no identities in the appliance's managed identity sources. That is, the users in the internal identity store have a unique identity for the appliance based on their email addresses.
- ❒ If a user has an identity in any of the appliance's managed identity sources, while the user is logged in through his identity provider account, he cannot access applications based on the entitlements associated with the managed user account.
- ❒ Account provisioning is not supported for the users in the SAML2 In unmanaged internal identity store. Because these users do not have a workforceID, they cannot be provisioned for or access the SaaS applications that depend on the workforceID attribute for authentication, such as Google Apps and Salesforce.

  For more information, see Section 2.4.1, "Requirements for Provisioning," on page 18.
- ❒ Users in a SAML2 In internal identity store are not supported in administration roles for the appliance because their passwords are not stored in the local identity store on the appliance.

3.5.3 SAML2 In Requirements for the Identity Provider

To create a custom SAML 2.0 Inbound connector for an identity provider, ensure that the identity provider meets the following requirements:

❒ Supports identity federation using the SAML 2.0 protocol.

For more information about SAML, see the OASIS website (https://wiki.oasis-open.org/security/FrontPage).

❒ Supports the SAML web browser single sign-on profile, with the Redirect and POST bindings for service-provider-initiated SSO, and the POST binding for identity-provider-initiated SSO.

❒ Provides a capability in the application’s administration console that allows the customer to enable and configure SAML SSO.

When you configure the SAML2 In connector, the **Federation Instructions** provide the information that you will need to set up the federation for CloudAccess in the identity provider. This information includes the metadata, a signing certificate for the appliance, the field values to use, and other guidance.

The SAML 2.0 metadata for the appliance does not contain SAML 2.0 service provider information by default. You must configure at least one instance of a SAML2 In identity source before the appliance publishes service provider information in its metadata. To verify that a SAML2 In identity source is properly configured, open the appliance’s metadata and search for the `SPSSODescriptor` tag.

For information about importing and configuring the SAML2 In connector, see Section 3.9, “Importing and Configuring Custom Connectors,” on page 38.

❒ Provides technical documents that describe SAML federation requirements, metadata, and assertions.

❒ Provides an Email attribute for every user. You will map this attribute to the SAML2 In NameID attribute.

The SAML2 In identity source uses the value mapped to the NameID attribute in an assertion to uniquely identify the user. For more information about the role of email addresses for SAML2 In users, see Section 3.5.2, “Requirements for Using SAML2 In Identity Sources,” on page 33.

3.5.4 Planning for a SAML2 In Connector

Before you attempt to create the connector, you must collect information about the originating identity provider. Ask the identity provider vendors the following types of questions to gather the required information:

- What does your SAML assertion look like?
- Do you have a SAML metadata document? What fields, if any, are customer-specific?
- Does your service support the SAML single logout protocol?
- What are the required configuration steps in your application to set up federation?
- What is the information that you provide to customers when they are setting up federation?

NOTE: You can use a worksheet to organize the information. See “Worksheet for SAML In Custom Connectors”.

3.5.5 Creating a SAML2 In Connector for an Identity Provider

A SAML2 In connector template consists of multiple components for federation, metadata, and assertion information.

To create a custom connector template:

1 Log in as an administrator to the Access Connector Toolkit.

2 Click **New** > **SAML2 In**.

The connector **Type** is `SAML2 In`. The **Type Name** is `Generic SAML2 In Connector`.

3 On the **Template** tab, complete the following information:

- Template properties
- Whether the service provider requires a signing certificate
- Federation instructions for the service provider
- New settings that need to be collected on the Configuration page of the connector

4 Click the **Metadata** tab, then use one of the following methods to specify the metadata:

- Select **Request**, then specify the source URL to retrieve the metadata.
- Complete the fields to manually generate the metadata.
- Import the values from a file or URL, and modify them for your deployment environment.

5 Click the **Assertion** tab, then define the properties and attributes required for the security token.

5a On the **Properties** subtab, specify the properties for the assertion.

5b On the **Attributes** subtab, click **Predefined**, click the identity attribute, modify the definition if needed, then click **Save**.

Set **NameID** to the identity provider attribute that contains the user’s email address.

If a predefined option does not exist, use **New** to define it.

5c (Conditional) If the service provider requires other identity attributes for an assertion, repeat Step 5b to map the WS-Federation attribute to an attribute in your identity source.

6 Click **Save** to save the new connector template.

7 Proceed to Section 3.8, “Exporting a Connector Template,” on page 38 to finish creating the new connector.

3.6 Creating a Basic SSO Connector Template

A connector for basic single sign-on uses HTML Forms to populate the authentication information. To create a custom connector for Basic SSO, you must define the HTML form for the desired application.

- Section 3.6.1, “Basic SSO Requirements,” on page 35
- Section 3.6.2, “Planning for Basic SSO,” on page 36
- Section 3.6.3, “Creating a Basic SSO Connector Template for a Web Service,” on page 36

3.6.1 Basic SSO Requirements

Note the following requirements before you create a custom connector for Basic SSO:

❒ The application or web service must support HTML Forms.

For more information, see www.w3.org (http://www.w3.org/TR/html401/interact/forms.html).

- ❒ The connector supports user access to destination websites only through Chrome, Internet Explorer, and Firefox web browsers running on a desktop or laptop computer. The connector works with the Basic SSO extension to securely collect, store, retrieve, and replay users' credentials for their destination websites.

  The MobileAccess app supports the secure retrieval and replay of previously stored credentials for websites that users access through the landing page on supported mobile devices.

- ❒ A user must install the Basic SSO extension in a supported browser one time on each desktop or laptop they use to access the Basic SSO websites.

  For Chrome, the extension is available for free from the Google Play Store. If it is not installed when the user accesses the application through CloudAccess, CloudAccess prompts the user to go to the Google Play Store and install it. The extension is added to the Chrome Extensions list, with the following permissions:

  - Access your data on all websites
  - Access your tabs and browsing activity

  For Firefox, the extension is available through Add-ons (https://addons.mozilla.org/en-US/firefox/). The Firefox extension behaves the same way as the Chrome extension.

  For Internet Explorer, CloudAccess prompts the user to install the Basic SSO extension from a location on the appliance.

3.6.2 Planning for Basic SSO

Before you attempt to create the connector, you must collect information about the format of the HTML form on the login page of the web service or application. For example:

- What is the domain URL for the web service or application?
- What is the login page for the web service or application?
- What is the form ID or name for the user name?
- What is the form ID or name for the user password?
- What input type is used for the form (button, image, string)?

NOTE: You can use a worksheet to organize the information. See "Worksheet for Basic SSO Custom Connectors".

3.6.3 Creating a Basic SSO Connector Template for a Web Service

A Basic SSO connector template consists of multiple components. CloudAccess contains an interface that allows you to create the components in one place.

To create a connector template for Basic SSO:

1 Log in as a CloudAccess administrator to the Access Connector Toolkit at:

   `https://`*`appliance_dns_name`*`/css/toolkit`

2 Click **New** > **Basic SSO**.

   The connector **Type** is `Basic SSO`. The **Type Name** is `Generic Basic SSO Connector`.

3. On the **Template** tab, complete the template properties:
   - The unique name for the template file (target name). This name cannot include spaces or special characters.
   - A brief description of the connector.
   - A 3-digit version number (ex: 1.0.0).
   - A custom graphic to use for the icon that represents the connector on the Admin page.
4. (Conditional) If you need to specify a variable for the destination URL, under **Settings**, click **New**, then define the variable settings.

   For example, some websites require an assigned keyword or organization name for the URL assigned to your company's account.
5. Click the **Forms** tab and create the form for the Basic SSO connector.

   The **Forms** tab allows you to define the HTML form for the appropriate application and platform. You define the HTML form fields that are required to populate the form correctly. Use the information from w3.org (http://www.w3.org/TR/html401/interact/forms.html) to create the form.

   Sometimes the application's HTML form is different for the desktop application or the mobile application. This means you must create multiple forms for the application. You can use the same fields for all three platforms or define a unique form for each platform.
6. Click **Save** to save the new connector template.
7. Proceed to Section 3.8, "Exporting a Connector Template," on page 38 to finish creating the new connector.

3.7 Modifying a Connector

You can modify the definition information for a connector by importing it in the Access Connector Toolkit. For example, you can import an existing connector to update its definition to the latest features available for connectors.

1. Obtain a copy of the connector's ZIP file.
2. Log in as a CloudAccess administrator to the Access Connector Toolkit at:

   ```
   https://appliance_dns_name/css/toolkit
   ```
3. Click **Import**, browse to select the connector's ZIP file, then click **OK**.

   The connector appears in the list of connector templates.
4. Click the **Edit** icon next to the **Display Name** for the connector template to open it in the Edit Connector Template window.
5. Modify the connector template settings as desired.
6. Click **Save** to apply the changes.
7. Click the **Export** icon next to the **Display Name** for the connector template.
8. Save the ZIP file for use on this or another CloudAccess system.
9. Proceed to Section 3.9, "Importing and Configuring Custom Connectors," on page 38.

3.8 Exporting a Connector Template

After you create a connector template, you must use the Access Connector Toolkit to export it in a compressed ZIP file that you can import to any CloudAccess system. You then import the connector template in the CloudAccess administration console to make it available in the **Applications** palette.

To export the connector template:

1. Log in as a CloudAccess administrator to the Access Connector Toolkit at:

   `https://`*`appliance_dns_name`*`/css/toolkit`

2. Click the **Export** icon next to the **Display Name** for the connector template.
3. Save the ZIP file for use on this or another CloudAccess system.
4. Proceed to Section 3.9, "Importing and Configuring Custom Connectors," on page 38.

3.9 Importing and Configuring Custom Connectors

CloudAccess allows you to import and configure custom connectors that you create with the Access Connector Toolkit, or that are created for you by Technical Support or partners.

After you export a custom connector, you must import its ZIP file to CloudAccess to make it available in the **Applications** palette of the administration console. Thereafter, you can enable and manage the connector as you do the connectors for applications that shipped with the appliance. The custom connector might require additional configuration, depending on the single sign-on method you use.

The destination application might also require additional configuration, depending on the application and the federation method. The destination applications for connectors for Basic SSO do not require additional configuration.

- Section 3.9.1, "SAML2 and WS-Fed Custom Connectors," on page 38
- Section 3.9.2, "SAML2 In Custom Connectors," on page 39
- Section 3.9.3, "Basic SSO Custom Connectors," on page 40

3.9.1 SAML2 and WS-Fed Custom Connectors

To import and configure a custom connector for SAML2 and WS-Federation:

1. Log in as an administrator to the CloudAccess administration console:

   `https://`*`appliance_dns_name`*`/appliance/index.html`

2. Import the custom connector to the **Applications** palette.

   2a Copy the custom connector ZIP file to the computer where you administer CloudAccess.

   2b On the Admin page, click the **Tools** icon on the toolbar, then click **Import connector template**.

   2c Browse to and select the custom connector ZIP file, then click **Import**.

   The connector appears in the **Applications** palette.

3. Drag the new custom connector from the **Applications** palette to the **Applications** panel.
4. Complete the connector settings on the **Configuration** tab.

   The steps to configure the connector are determined by the information you added to the connector template.

5. Expand the **Federation Instructions**, then copy and paste the instructions into a text file to use when you configure the destination application.

   NOTE: You must use a text editor that does not introduce hard returns or additional white space. For example, use Notepad instead of Wordpad.

6. Click the **Appmarks** tab, then review and edit the default settings for the appmark.

   For more information, see Chapter 5, “Configuring Appmarks for Connectors,” on page 45.

7. Click **OK** to save the configuration.
8. On the Admin page, click **Apply** to commit the changes to the appliance.
9. Wait until the configuration changes have been applied on each node of the CloudAccess cluster.
10. Log in to the service provider as the account administrator, then configure the federation for CloudAccess in the application’s administration console.

    Use the information from the **Federation Instructions** in Step 5 to complete the setup.

    NOTE: When you copy the appliance’s signing certificate, ensure that you include all leading and trailing hyphens in the certificate’s Begin and End tags.

11. In the CloudAccess administration console, click **Policy** on the toolbar, then perform policy mapping to specify entitlements for the SAML 2.0 Inbound users to the service provider application.

    For more information, see “Mapping Authorizations” in the *CloudAccess Installation and Configuration Guide*.

12. After you complete the configuration, users can log in through CloudAccess to single sign-on to the service provider’s system. The CloudAccess login page URL is:

    ```
    https://appliance_dns_name
    ```

3.9.2 SAML2 In Custom Connectors

Before you begin, ensure that you understand the trust policy settings for the SAML2 In identity sources. For more information, see Section 3.5.1, “Understanding SAML2 In Identity Sources,” on page 32 and Section 3.5.2, “Requirements for Using SAML2 In Identity Sources,” on page 33.

To import and configure a custom connector for SAML2 In as an identity source:

1. Log in as an administrator to the CloudAccess administration console:

   ```
   https://appliance_dns_name/appliance/index.html
   ```

2. Import the custom SAML2 In connector to the **Applications** palette.

   2a Copy the custom connector ZIP file to the computer where you administer CloudAccess.

   2b On the Admin page, click the **Tools** icon on the toolbar, then click **Import connector template**.

   2c Browse to and select the custom connector ZIP file, then click **Import**.

   The connector appears in the **Identity Sources** palette.

3. Drag the new custom connector from the **Identity Sources** palette to the **Identity Sources** panel.
4. Complete the connector settings on the **Configuration** tab.

The steps to configure the connector are determined by the information you added to the connector template.

5. Under **Assertion Attribute Mappings**, map the SAML Assertion attributes to the appropriate attributes in your identity source.
6. Under **Trust policy for user identities in assertions**, configure the preferred action to take when the appliance receives an assertion from the identity provider:

   - **Allow access for unknown users:** Creates unique user objects in an unmanaged internal identity store for the identity provider.

     For more information, see Section 3.5.2, "Requirements for Using SAML2 In Identity Sources," on page 33.

   - **Allow access for known users:** Matches user objects in managed identity sources.

   For more information about unknown and known users, see Section 3.5.1, "Understanding SAML2 In Identity Sources," on page 32.
7. Expand the **Federation Instructions**, then copy and paste the instructions into a text file to use when you configure the originating identity provider.

   NOTE: You must use a text editor that does not introduce hard returns or additional white space. For example, use Notepad instead of Wordpad.
8. Click **OK** to save the configuration.
9. On the Admin page, click **Apply** to commit the changes to the appliance.
10. Wait until the configuration changes have been applied on each node of the CloudAccess cluster.
11. The appliance now acts as a SAML 2.0 service provider for the specified identity provider. The appliance SAML 2.0 metadata should now include the `SPSSODescriptor` section. Use this information to configure the identity provider for SAML 2.0 Inbound federation with the appliance.
12. Log in to the originating identity provider as the account administrator, then configure the SAML 2.0 Inbound federation for CloudAccess in the provider's administration console.

    To complete the setup, use the information from the **Federation Instructions** in Step 7 and the `SPSSODescriptor` from Step 11.

    NOTE: When you copy the appliance's signing certificate, ensure that you include all leading and trailing hyphens in the certificate's Begin and End tags.
13. (Conditional) If you enabled access for unknown users, you must configure entitlements for the users that will be added to the SAML2 In internal data store. In the CloudAccess administration console, click **Policy** on the toolbar, then perform policy mapping to specify entitlements for the SAML2 In users to the appropriate applications.

    For more information, see "Mapping Authorizations" in the *CloudAccess Installation and Configuration Guide*.
14. The appliance login page provides a link to the login page of the SAML 2.0 identity provider, located to the left of the user name and password login options. The SAML 2.0 users log in through the identity provider to gain access to the appliance landing page.

3.9.3 Basic SSO Custom Connectors

To import and configure a custom connector for Basic SSO:

1. Log in as an administrator to the CloudAccess administration console:

```
https://appliance_dns_name/appliance/index.html
```

2 Import the custom connector for Basic SSO to the **Applications** palette.

   2a Copy the custom connector ZIP file to the computer where you administer CloudAccess.

   2b Click the **Tools** icon on the console toolbar, then click **Import Connector Template**.

   2c Browse to and select the ZIP file for the connector template you want to import, then click **Import**.

   The imported connector appears in the **Applications** palette.

3 Drag the new custom connector from the **Applications** palette to the **Applications** panel.

4 On the Configuration page, you can modify the display name, set custom settings as required, and view information about the connector’s application.

5 Click the **Appmarks** tab, then review the default settings for the appmark.

   Public access is enabled automatically. If you disable public access, the appmark does not appear on the landing page until you map authorizations to set entitlements for user roles (groups).

6 Click **OK** to save the configuration.

7 On the Admin page, click **Apply** to commit the changes to the appliance.

8 Wait until the configuration changes have been applied on each node of the CloudAccess cluster.

9 (Conditional) If Public access is disabled, perform policy mapping to specify entitlements for identity source roles (groups).

   For more information, see “Mapping Authorizations” in the *CloudAccess Installation and Configuration Guide*.

10 After you complete the configuration, users can log in through CloudAccess to access the application. The CloudAccess login page URL is:

```
https://appliance_dns_name
```

4 Importing Connectors from the Application Connector Catalog

Not all connectors that are available for CloudAccess ship with the appliance. To use additional (non-custom) Basic SSO and Single Sign-On connectors, you must import them from the Application Connector Catalog.

NOTE: For information about importing custom connectors created in the Application Connector Toolkit, see Section 3.8, “Exporting a Connector Template,” on page 38 and Section 3.9, “Importing and Configuring Custom Connectors,” on page 38.

To import connectors from the Application Connector Catalog:

1 Log in as an administrator to the CloudAccess administration console:

```
https://appliance_dns_name/appliance/index.html
```

2 On the Applications palette, click **Add Application** (the plus (**+**) icon) to open the Application Connector Catalog.

3 Locate and select the appropriate connector in the catalog, then click **Import**.

> **NOTE:** Applications are listed in alphabetical order in the catalog, but might not be in the location you expect. For example, the WebEx connector is listed as Cisco WebEx. Enter the application name in the **Search** field if you have trouble locating a connector by browsing.

4 Repeat as needed to import additional connectors.

Each connector import occurs as a separate request. You do not need to wait for the import of a connector to complete before starting another import.

5 When the connector import succeeds, a green check mark icon appears next to the application connector in the catalog, and the **Applications** palette displays the connector icon.

If an error occurs during an import, a message appears beneath the connector icon in the catalog. Click **Import** to try again. You can view details about import errors in the appliance log files.

6 Drag the application icon from the **Applications** palette to the **Applications** panel.

After you import the connector, you must configure the connector settings in CloudAccess. For more information, see the following sections:

- Section 8.3, “Importing and Configuring a Connector for Basic SSO,” on page 60
- Section 13.3, “Configuring Single Sign-On Connectors,” on page 94

5 Configuring Appmarks for Connectors

Appmarks are essentially bookmarks for applications. After you configure a connector for an application, you configure one or more appmarks to enable users to access the application in different ways. After a user logs in to CloudAccess, users see the appmarks that they are entitled to see on the landing page, according to the application settings for public access or policy mappings for the application to identity source roles (groups).

Use the information in the following sections to help you understand and configure appmarks:

- Section 5.1, “Understanding Appmarks,” on page 45
- Section 5.2, “Configuring an Appmark for the Desktop Browser or Mobile Device,” on page 47
- Section 5.3, “Creating Multiple Appmarks for an Application,” on page 48
- Section 5.4, “Using Appmark Variables,” on page 49
- Section 5.5, “Policy Mapping for Non-Public Appmarks,” on page 49

5.1 Understanding Appmarks

You can configure appmarks for any proxy connector, SaaS connector, or SSO connector. You can even configure multiple appmarks for the same connector. For example, you might want to have several appmarks for the various Office 365 applications so users can easily identify them. The connector for Google Apps includes default appmarks for Calendar, Drive, and Mail applications. You can copy an existing appmark to create a new one.

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, after the appliance is finished applying your change, the user must do a refresh on the Applications page on the mobile device before the appmark appears.

NOTE: Appmarks for proxy and SSO connectors have no access control associated with them. If users know how to get to a service, they can access the service. Appmarks just add convenience to the user experience.

You configure appmarks on the Appmarks tab in the configuration window for the connector. On the Appmarks tab next to the name of the appmark in the blue bar are several icons for renaming, copying, disabling, or deleting the appmark. Use the mouseover text to identify the icon you want to use. You can view and edit appmark configuration options by clicking the blue bar or the plus sign (+) icon. The following appmark options are available:

Reset

This check box restores the Appmarks tab to the default settings for the connector. Consider using this option if you have configured custom connectors that are not working as expected. Click **OK** and apply the changes to the appliance to see the default appmark settings.

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 should create a copy of the appmark and change the name. For more information, see Section 5.3, “Creating Multiple Appmarks for an Application,” on page 48.

Public

This option is available only for appmarks configured for Bookmarks, OAuth2 Resources, and SSO-only type connectors. Public access is disabled by default for all connectors except connectors for Basic SSO. If you select the **Public** option, all users can see and use the appmark. If you deselect the **Public** option, no users can see the appmark until it is mapped to desired identity source roles (groups) in Policy Mapping.

Desktop browser

Enables the appmark to be visible on the CloudAccess landing page.

Initiate login at

Specifies whether the URL of the appmark on the landing page is the identity provider-initiated type or the service provider-initiated type. This option appears only for the full provisioning connectors (Google Apps, Salesforce, ServiceNow, and Office 365) and the SSO-only connectors, such as Box or Accellion.

URL

The URL that is to be used for the appmark. There are some replacement values that you can use. For more information, see Section 5.4, “Using Appmark Variables,” on page 49.

Icon

The icon that appears on the landing page. Within the same appmark, you can use different icons for the landing page and for mobile devices. You can use a different custom icon for each connector to improve usability for users.

iOS devices

Enables the appmark to appear on supported iOS mobile devices in the MobileAccess app.

Android devices

Enables the appmark to appear on supported Android mobile devices in the MobileAccess app.

Launch with

Specifies how to launch the application on the mobile device. Options include the following:

- **Safari:** When the user opens the MobileAccess app on the mobile device and taps the appmark, the app launches Safari and directs it to the application.
- **Chrome:** When the user opens the MobileAccess app on the mobile device and taps the appmark, the 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 to install it.
- **Internal viewer:** When the user opens the MobileAccess app on the mobile device and taps the appmark, the 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 the tool in the appliance) 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.

- **Native application:** Use this option specifically for mobile apps. When the user opens the MobileAccess app on the mobile device and taps the appmark, MobileAccess opens the mobile app itself.

Launch URL

Use for the **Native application** option. This is the URL such as `fb://profile` that will launch another application installed on the device.

App installer URL

(Optional) You can use this option if you selected the **Native application** option. This is the URL to install the application if it is missing on the mobile device.

URL

The URL that is to be used for the appmark. This can be different from the desktop URL if there is a mobile-specific version of the page.

Icon

The icon that represents the application in the MobileAccess app. Appmark icons for mobile devices should be in `.png` file format and ideally 72 x 72 pixels to ensure they display correctly. Square icons size well on mobile devices. Each icon should convey a good visual image of the application it represents.

5.2 Configuring an Appmark for the Desktop Browser or Mobile Device

After you have configured a connector for a proxy, SaaS, or SSO application, you can configure an appmark to simplify access to that application from the user's landing page or from a mobile device, or both.

To configure an appmark:

1. Log in with an appliance administrator account to the Admin page at

   https://*appliance_dns_name*/appliance/index.html

2. (Conditional) If you have not already configured the connector for the application, drag it from the **Applications** palette to the **Applications** panel.
3. Click the configured connector on the **Applications** panel and click **Configure**.
4. (Conditional) If you have not already configured the connector, provide the appropriate information on the **Configuration** tab. The required information varies depending on the connector.
5. Click the **Appmarks** tab.
6. Click the plus (**+**) sign next to the default created appmark.
7. (Conditional) Select the **Public** check box if you want the appmark to appear for all users, regardless of their entitlement to the application.

8 (Conditional) If you want the appmark to be available on the user's landing page, select the **Desktop browser** check box and complete the following steps:

   8a (Conditional) If it is applicable to the connector, select the appropriate option from the **Initiate login at** list.

   8b Leave the default value in the **URL** field.

   8c (Optional) If you want to provide your own icon for the appmark, click the **X** on the **Icon** line to delete the default icon. Then browse to and select a `.png` file to represent the application on the browser's landing page.

9 (Conditional) If you want the appmark to be available on the user's mobile device, select the **iOS devices** or **Android devices** check box and complete the following steps.

   9a Select an option from the **Launch with** list to specify how you want users to access the application on their mobile device. For more information about the available options, see Section 5.1, "Understanding Appmarks," on page 45.

   9b (Optional) If you want to provide your own icon for the appmark, click the **X** on the **Icon** line to delete the default icon. Then browse to and select a `.png` file to represent the application on the mobile device. You can use different icons for the landing page and mobile devices.

10 Click **OK**, then click **Apply**.

The appliance reconfigures with the new change. After this process has completed, users who enter the appliance URL are redirected to a login page. They enter their user name and password and are presented with a landing page containing the appmark icon that links to the application.

5.3 Creating Multiple Appmarks for an Application

Application connectors can have multiple appmarks. For example, you might create several appmarks for different Office 365 or Google Apps applications. You can create a new appmark from scratch, or you can copy an existing appmark to save time, especially if you want to create several appmarks and just change one or two options on each one. This procedure assumes you have already configured the connector.

To create a new appmark for a connector:

1 Log in with an appliance administrator account to the Admin page at

```
https://appliance_dns_name/appliance/index.html
```

2 Click the configured connector on the **Applications** panel, then click **Configure**.

3 Click the **Appmarks** tab, then do one of the following:

   - Click **New**
   - Click the **Copy** icon next to the existing appmark name

4 (Conditional) If you are copying an existing appmark, the **Name** field is pre-populated with `COPY_$(DisplayName)`. You have several 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.
   - You can edit the display name by selecting from available variables. Type `${` at the end of the field, then select a variable from the list. For more information about the available variables, see Section 5.4, "Using Appmark Variables," on page 49.

5 Specify whether the application should be accessible from a desktop browser, a mobile device, or both, and complete the appropriate fields. For more information about available options, see Section 5.1, "Understanding Appmarks," on page 45.

6 Click **OK**, then click **Apply** to update the appliance.

5.4 Using Appmark Variables

Each connector has different configuration settings and variables, and some appmarks need to contain information from the connector configuration to be useful. When you configure a connector, the Appmarks tab is automatically populated with one or more default appmarks, depending on the connector. The default settings contain some variables in the URL field.

You can use the variables that are available for a connector in the **Name** and **URL** fields if they are of the string type and have a value provided. To insert a variable, type `${` to display the available variables. Use the mouse or press the up/down arrow keys to select a variable. When you press the down arrow key, an additional box shows the resolved value. Press the up arrow key to close the resolved variables box. Some variables may not be resolvable until after you apply your changes on the appliance.

5.5 Policy Mapping for Non-Public Appmarks

Appmarks for proxy and SSO applications are intended only for display and convenience. They are not connected to any authorization policy or access control list (ACL). The SSO and proxy appmark URLs are still available to be used by anyone who knows the link in the URL field. However, selecting or deselecting the **Public** option when configuring an appmark determines whether the appmark actually appears for the users in a group. If you deselect the **Public** check box, the appmark is not available for users until you map the appmark to one or more groups in your configured identity source. After mapping is completed, users in those mapped groups can see the appmark on the landing page or mobile device.

The following procedure assumes that you have already configured an appmark and applied the change on the appliance.

To map an appmark to a group in your identity source:

1 Switch to the Policy page of the administration console.

2 On the left side, locate the identity source that has the desired group (listed as Role Name) from the list.

3 On the right side, select **Other Applications** from the list.

4 Select the Authorization Name of the appmark and drag it to a Role Name.

5 In the mapping window, there are no approvals for appmarks because there is no account provisioning in this process. Users who are included in the group are automatically approved. Click **OK** to continue.

Now when users who are in the mapped group do a refresh in the MobileAccess app or access the landing page, they see the new appmark icon. Users who are not in the mapped group do not see the icon.

6 Connector for Bookmarks

The connector for Bookmarks on the **Applications** palette enables you to create links to web applications that are accessible from the browser landing page or directly from the MobileAccess app on users' mobile devices.

You can also create links to other mobile applications from the MobileAccess app, though there is no single sign-on for these apps.

While there is no global list for these app URL schemes, you might find the following list helpful:

http://wiki.akosma.com/IPhone_URL_Schemes

6.1 Configuring the Connector for Bookmarks

The connector for Bookmarks is intended as a container for multiple appmarks for web applications. Configure the Bookmarks connector once, then configure as many appmarks as you need within the same Bookmarks connector so you do not clutter your Admin page.

To configure a bookmark connector:

1. Log in as an administrator to the CloudAccess administration console:

   `https://`*`appliance_dns_name`*`/appliance/index.html`

2. Drag the **Bookmark** connector from the **Applications** palette to the **Applications** panel.
3. Provide a display name for the bookmarked application. The display name appears on the Admin page of the administration console and in the MobileAccess app.
4. Click the **Appmarks** tab.
5. Click the plus (**+**) sign next to the default created appmark.
6. Rename the appmark to correspond to the bookmark URL.
7. (Conditional) Select the **Public** check box if you want the appmark to appear for all users, regardless of their entitlement to the application.
8. (Conditional) If you want users to be able to access the bookmarked application from their desktop browser landing page, select **Desktop browser** and complete the following steps:

   8a In the **URL** field, change the default value to the URL of the bookmarked application.

   8b Click **X** next to the default icon to delete it, then browse to and select a `.png` file to represent the application on the browser landing page.

9. (Conditional) If you want users to be able to access the bookmarked application from their mobile devices, select **iOS devices** or **Android devices** and specify the appropriate options as follows:

   9a From the **Launch with** list, select the viewer in which the application should appear on mobile devices: Safari, Chrome, or an internal viewer.

NOTE: If the URL for the appmark points to a destination web server that uses a non-public signing certificate for SSL, configure the appmark to open in a Safari or Chrome browser. With an internal viewer, users will receive a certificate error and their mobile devices cannot display the page.

9b Leave the **Launch URL** and **App installer URL** fields blank.

9c In the **URL** field, type the same URL that you provided in step 7a or type a mobile-specific URL.

9d Click **X** next to the default icon to delete it, then browse to and select a `.png` file to represent the application on the mobile device.

10 (Conditional) If you want to use the Bookmark connector to link to other mobile applications from the MobileAccess app, select **iOS devices** or **Android devices** and specify the following options:

10a From the **Launch with** list, select **Native application**.

10b In the **Launch URL** field, enter the mobile app URL scheme. For example, `fb://profile`.

10c (Optional) In the **App installer URL** field, type the URL to install the application if it has not already been installed on the mobile device.

10d Click **X** next to the default icon to delete it, then browse to and select a `.png` file to represent the bookmarked application.

11 Click **OK**.

12 On the Admin page, click **Apply** to commit the changes to the appliance.

13 Wait until the configuration changes have been applied on each node of the CloudAccess cluster.

To see the application on their mobile devices, users must perform a refresh on the Applications page in the MobileAccess app. How users access the bookmark appmark depends on how you configured the **Launch with** option.

7 Connector for OAuth2 Resources

The connector for OAuth 2 Resources provides simple authenticated access to a web service through CloudAccess. The connector allows CloudAccess to authenticate a user against your identity sources and to provide protected access to a destination web service.

The connector for OAuth2 Resources offers a simple authentication method as an alternative to federated single sign-on connectors that use SAML 2.0 or WS-Federation protocols. Protocols for federated access management provide a robust trust and security model that is an open standard and widely used. However, it does require the protocol's code to be installed on the protected services. Consider using the connector for OAuth2 Resources for smaller services that do not require the full security and trust that SAML or WS-Federation provides, and just need a simple method to validate and get identity information from a trusted source (the CloudAccess identity provider in this case).

By implementing the open standard OAuth 2.0 protocol, the connector for OAuth2 Resources behaves as an OAuth2 Authorization Server and Resource Server using the Authorization Code flow as detailed in the OAuth 2.0 Authorization Framework document at http://tools.ietf.org/html/rfc6749#section-4.1.

Using this connector, the CloudAccess appliance provides user authentication and all OAuth2 token creation and validation for access to a protected resource.

NOTE: The OAuth2 Resources connector provides SP-initiated authentication. It does not have an IDP-initiated mode.

Use the information in the following sections to configure a connector for OAuth2 Resources:

- Section 7.1, "Configuring the OAuth2 Client Application," on page 53
- Section 7.2, "Configuring the Connector for OAuth2 Resources," on page 54
- Section 7.3, "Supported OpenID Connect Schema," on page 55

7.1 Configuring the OAuth2 Client Application

When you configure the connector for OAuth2 Resources on CloudAccess, the Client ID, Client Secret, and OAuth Endpoint URLs are created automatically. This information must then be used to configure the OAuth2 client application. All configuration activities at the OAuth2 client application are out of band.

Enforcement of authorization or access control beyond the initial authentication and token creation process is the responsibility of the OAuth client application, since the OAuth Resources connector does not currently support policy mapping in CloudAccess.

For information about configuring the OAuth client application, refer to your OAuth client application documentation.

7.2 Configuring the Connector for OAuth2 Resources

You can configure instances of the OAuth2 Resources connector in one of the following ways:

- An instance of the connector per OAuth client application. This is the simplest method conceptually and matches how SAML connectors are used.
- Multiple OAuth client applications all configured within a single instance of the OAuth2 Resources connector. This means that all OAuth2 client applications would use the same schema (OpenID Connect or native), and would use the same Client ID and Client Secret. This configuration is simple to configure and maintain, but care should be taken to include only clients of the same trust level in a connector instance. Because all clients share the same client ID and secret, if one of the clients is compromised in any way, they are all compromised. Any of them could also masquerade as another client in some cases.

(Optional) For each OAuth client application, you can manually create appmarks so the CloudAccess landing page shows an icon for connection to the OAuth2 client application. Appmarks should be configured to point to the URL of the OAuth2 client application that will start the OAuth2 authentication process.

To configure the connector for OAuth2 Resources:

1. Log in as an administrator to the CloudAccess administration console:

   `https://`*`appliance_dns_name`*`/appliance/index.html`

2. Drag the **OAuth Resources** connector from the **Applications** palette to the **Applications** panel.
3. On the **Configuration** tab, provide the following information:
   - **Display name**: Clearly identify the connector on the Admin page of the console.
   - **Schema**: Specify whether the attributes that CloudAccess sends to the OAuth client follow OpenID Connect standard naming or use the Native schema names defined internally on the appliance.
   - **Allowed OAuth Client URI(s)**: Specify the whole path or just the host name for the OAuth2 client application. Using only the host name allows all paths on that domain. Since OAuth2 depends on SSL as one of its core security mechanisms, HTTPS should always be specified. For more information about configuring redirect URIs, see the following document: http://tools.ietf.org/html/rfc6749#section-10.6.
   - **OAuth Details (Client ID and Client Secret)**: Use this information to configure the OAuth2 client application.
   - **OAuth Endpoints (Auth URL, Token URL, and Profile URL)**: Use this information to configure the OAuth2 client application.
4. Click the **Appmarks** tab, then review and edit the default settings for the appmark.
5. Click **OK** to save the configuration.
6. On the Admin page, click **Apply** to commit the changes to the appliance.
7. Wait until the configuration changes have been applied on each node of the CloudAccess cluster.
8. (Conditional) If Public access is disabled, click **Policy** in the toolbar, then perform policy mapping to specify entitlements for identity source roles (groups).

   For more information, see “Mapping Authorizations” in the *CloudAccess Installation and Configuration Guide*.

After the OAuth2 Resources connector and OAuth client application have been configured, end users can access the protected resource by browsing to the URL of the OAuth client application (by entering the URL directly into the browser, using a bookmark or the landing page appmark, and so forth). If the user is not already authenticated to the CloudAccess appliance, the browser is redirected to the CloudAccess login page and the user is prompted for login credentials. After a successful authentication or if the user is already authenticated to the appliance and is authorized to access the protected resource, the user gains access to the resource.

7.3 Supported OpenID Connect Schema

The OAuth Resources connector supports the OpenID Connect schema names listed in the following table.

Table 7-1 *OpenID Connect Schema*

Member	Type	Description
name	string	End user's full name in displayable form including all name parts, possibly including titles and suffixes, ordered according to the user's locale and preferences.
given_name	string	Given name(s) or first name(s) of the end user. Note that in some cultures, people can have multiple given names; all can be present, with the names being separated by space characters.
family_name	string	Surname(s) or last name(s) of the end user. Note that in some cultures, people can have multiple family names or no family name; all can be present, with the names being separated by space characters.
middle_name	string	Middle name(s) of the end user. Note that in some cultures, people can have multiple middle names; all can be present, with the names being separated by space characters. Also note that in some cultures, middle names are not used.
preferred_username	string	Shorthand name that the end user wishes to be referred to at the RP, such as janedoe or j.doe. This value *may* be any valid JSON string including special characters such as @, /, or whitespace. This value *must not* be relied upon to be unique by the RP. (See Section 2.5.3 (http://openid.net/specs/openid-connect-basic-1_0-28.html#claim.stability) of the OpenID Connect Basic Client Profile 1.0 document.)
picture	string	URL of the end user's profile picture. This URL *must* refer to an image file (for example, a PNG, JPEG, or GIF image file), rather than to a Web page containing an image. Note that this URL *should* specifically reference a profile photo of the end user suitable for displaying when describing the end user, rather than an arbitrary photo taken by the end user.
email	string	end user's preferred email address. Its value *must* conform to the RFC 5322 (http://openid.net/specs/openid-connect-basic-1_0-28.html#RFC5322) addr-spec syntax. This value *must not* be relied upon to be unique by the RP, as discussed in Section 2.5.3 (http://openid.net/specs/openid-connect-basic-1_0-28.html#claim.stability) of the OpenID Connect Basic Client Profile 1.0 document.

Member	Type	Description
gender	string	End user's gender. Values defined by this specification are female and male. Other values *may* be used when neither of the defined values is applicable.
birthdate	string	End user's birthday, represented as an ISO 8601:2004 (http://openid.net/specs/openid-connect-basic-1_0-28.html#ISO8601-2004)[ISO8601-2004] YYYY-MM-DD format. The year *may* be 0000, indicating that it is omitted. To represent only the year, YYYY format is allowed. Note that depending on the underlying platform's date related function, providing just year can result in varying month and day, so the implementers need to take this factor into account to correctly process the dates.
locale	string	End user’s locale, represented as a BCP47 (http://openid.net/specs/openid-connect-basic-1_0-28.html#RFC5646) [RFC5646] language tag. This is typically an ISO 639-1 Alpha-2 (http://openid.net/specs/openid-connect-basic-1_0-28.html#ISO3166-1) [ISO639 1] language code in lowercase and an ISO 3166-1 Alpha-2 (http://openid.net/specs/openid-connect-basic-1_0-28.html#ISO3166-1) [ISO3166 1] country code in uppercase, separated by a dash. For example, en-US or fr-CA. As a compatibility note, some implementations have used an underscore as the separator rather than a dash, for example, en_US; Implementations *may* choose to accept this locale syntax as well.
phone_number	string	End user's preferred telephone number. E.164 (http://openid.net/specs/openid-connect-basic-1_0-28.html#E.164) [E.164] is *recommended* as the format of this Claim, for example, +1 (425) 555-1212 or +56 (2) 687 2400. If the phone number contains an extension, it is *recommended* that the extension be represented using the RFC 3966 (http://openid.net/specs/openid-connect-basic-1_0-28.html#RFC3966) [RFC3966] extension syntax, for example, +1 (604) 555-1234;ext=5678.

8 Connectors for Basic SSO

Each connector for Basic Single Sign-On (SSO) provides forms-based single sign-on to an application through CloudAccess. It meets the specific interactive and content requirements for logging in to the application. A connector works with the Basic SSO extension for supported browsers to securely collect, store, retrieve, and replay the user’s authentication information for that application. For information about how CloudAccess keeps the user’s credentials secure, see Section 8.2, “Understanding Basic Single Sign-On,” on page 58.

CloudAccess provides many connectors for Basic SSO that you can import from the Application Connector Catalog to your appliance. New and updated connectors are added to the catalog as they become available. However, your appliance must be connected to the internet for the Application Connector Catalog to work. Ensure that you have port 80 open on your firewall for communication to the Application Connector Catalog for the latest connectors.

IMPORTANT: Please contact Technical Support (https://www.netiq.com/support/) if a connector for Basic SSO is not yet available for the forms-based authentication websites that your users access. This helps us to define requirements and set priorities for future connectors for Basic SSO.

You can also create your own custom connectors for Basic SSO with the Access Connector Toolkit. For more information, see Chapter 3, “Creating Custom Connectors,” on page 27.

You must use the CloudAccess administration console to import and enable the connectors for Basic SSO that you want to make available to your users. The connector enables public access by default to give access to all users. You can alternatively map authorization policies to grant access to select groups of users.

Use the information in the following sections to configure a connector for Basic SSO:

- Section 8.1, “Requirements for Using Basic SSO with Applications,” on page 57
- Section 8.2, “Understanding Basic Single Sign-On,” on page 58
- Section 8.3, “Importing and Configuring a Connector for Basic SSO,” on page 60
- Section 8.4, “Troubleshooting Basic Single Sign-On,” on page 61

8.1 Requirements for Using Basic SSO with Applications

❒ Connectors for Basic SSO work with applications that require forms-based authentication for login. Typically, they have the following login requirements:

  - The application’s login page uses HTML Forms as the main point of interaction with the user.
  - The application requires the user’s password to be sent for logging in.
  - The application does not support using SAML 2.0 and WS-Federation protocols for federated trust relationships instead of sending passwords.

❒ The connectors for Basic SSO support user access to applications through supported web browsers running on a desktop or laptop computer. They work with the Basic SSO extension to securely collect, store, retrieve, and replay users’ credentials for their applications.

The connectors for Basic SSO support only the following browsers:

- Google Chrome (latest version)
- Mozilla Firefox (latest version)
- Internet Explorer 11

The MobileAccess app supports the secure retrieval and replay of previously stored credentials for applications that users access through the landing page on supported mobile devices.

For user access to applications on supported mobile devices, the MobileAccess app supports only the following mobile operating systems:

- iOS 9.*x* or later
- Android KitKat 4.4
- Android Lollipop 5.*x*

❐ Users must install the Basic SSO extension in a supported browser one time on each desktop or laptop they use to access the Basic SSO applications.

For Chrome, the extension is available for free from the Google Play Store. If it is not installed when the user accesses the application through CloudAccess, CloudAccess prompts the user to go to the Google Play Store and install it. The extension is added to the Chrome Extensions list, with the following permissions:

- Access your data on all websites
- Access your tabs and browsing activity

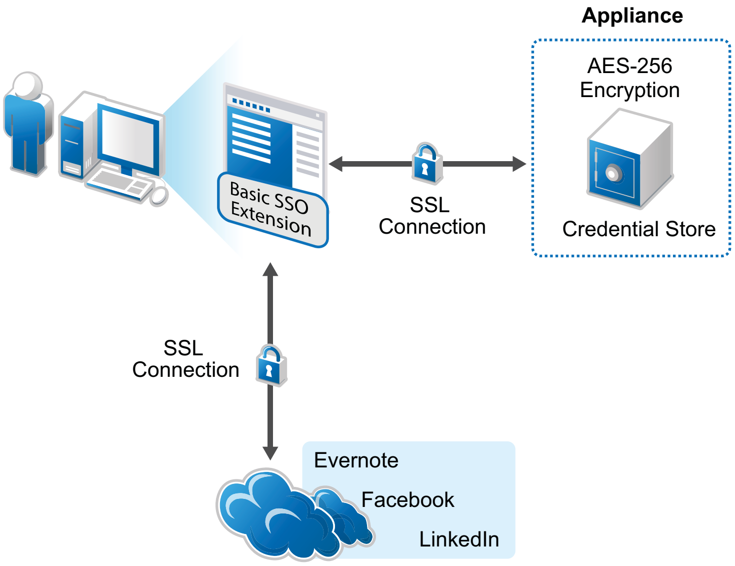
For Firefox, the extension is available through Add-ons (https://addons.mozilla.org/en-US/firefox/). The Firefox extension behaves the same way as the Chrome extension.

8.2 Understanding Basic Single Sign-On

Each connector for Basic SSO works with a browser extension for the destination website. It is designed specifically to collect and replay a user’s login credentials and metadata in a format that the site requires on its login page. After you configure connectors for Basic SSO, and create the appmark, the associated appmark appears in the user portal page.

CloudAccess protects user credentials through an SSL connection and AES-256 encryption on the appliance. Figure 8-1 depicts how CloudAccess stores the credentials securely.

Figure 8-1 Basic SSO Security



For users to access the Basic SSO connectors, they must install the appropriate Basic SSO extension or plugin for their browser or install the MobileAccess app. The following occurs the first time a user logs in to access a Basic SSO application:

1. The user logs in to CloudAccess using their CloudAccess credentials.
2. The user sees the available applications on the user portal page.
3. The user clicks the appropriate application icon.
4. If the Basic SSO extension or plugin for the browser is not installed on the computer, CloudAccess prompts the user to install it.
5. A new window opens for the login page of the application.
6. The user enters their user name and password for the application.

   The user must enter this separate user name and password once.
7. The extension or plugin captures the separate user name and password, then the extension or plugin sends the user name and password to CloudAccess over an SSL connection.
8. CloudAccess encrypts the user name and password with AES-256 encryption, and then stores the user name and password in the credential store that is part of CloudAccess.

   CloudAccess encrypts the user name and password with an encryption key that is unique per user.
9. CloudAccess sends the user to the application’s website over an SSL connection.

In subsequent CloudAccess sessions, the user can log in with the CloudAccess credentials and access the application without providing the additional credentials. CloudAccess securely retrieves and submits the user’s application login information for an automatic login on behalf of the user. Thus, the user has the experience of single sign-on.

The user must install the Basic SSO browser extension on each device where the user wants to access the application. CloudAccess automatically prompts the user to install the extension the first time the user accesses the application’s appmark from a different device, even if the user’s credentials for the application are available in the user store. The extension then retrieves and submits the user’s application login information from CloudAccess for an automatic login.

Typically, users have a different login user name and password for their individual accounts for each application. A user can have only one account per application. CloudAccess stores the user’s current credentials, but users still have the responsibility to maintain the credentials. The user uses the account management interface of the application to modify a password if it expires or is stolen.

If the user changes the user name or password to the account for the application, or if the user cancels the account, the user’s stored credentials are no longer valid. The automatic login fails, and the browser extension takes the user to the application’s login page where the user can log in with new credentials. CloudAccess removes the old credentials and stores the user’s new credentials for the application. CloudAccess uses the new credentials for subsequent logins.

NOTE: The MobileAccess app does not support the collection of a user’s credentials. It functions in replay mode only. If the user’s credentials are not in the store or if the stored credentials are invalid, the appmark on the landing page takes the mobile user to the application, but the user must enter the additional credentials.

8.3 Importing and Configuring a Connector for Basic SSO

You can import and configure as many connectors for Basic SSO as you need on your CloudAccess system.

To import and configure one or more connectors for Basic SSO:

1 Log in as an administrator to the CloudAccess administration console:

```
https://appliance_dns_name/appliance/index.html
```

2 Import the appropriate connector for Basic SSO to the **Applications** palette using either of the following methods:

- **Application Connector Catalog:** On the **Applications** palette, click **Add Application** (the **Plus** (**+**) icon) to open the Application Connector Catalog, browse or search to locate and select the appropriate connector, then click **Import**.

  You can import multiple connectors. Each connector import occurs as a separate request. You do not need to wait for the import of a connector to complete before starting another import.

  When the connector import succeeds, a green check mark icon appears next to the application connector in the catalog, and the **Applications** palette displays the connector icon.

  If an error occurs during an import, a message appears beneath the connector icon in the catalog. Click **Import** to try again. You can view details about import errors in the appliance log files.

- **Custom Connector:** You can manually import custom connectors for Basic SSO that you have created with the Access Connector Toolkit.

  1. Copy the custom connector ZIP file to the computer where you administer CloudAccess.
  2. Click the **Tools** icon on the console toolbar, then click **Import Connector Template**.
  3. Browse to and select the ZIP file for the connector template you want to import, then click **Import**.

  For more information, see Section 3.6, “Creating a Basic SSO Connector Template,” on page 35.

The imported connector appears in the **Applications** palette.

3. Drag the imported connector from the **Applications** palette to the **Applications** panel.
4. On the Configuration page, you can modify the display name, set custom settings as required, and view information about the connector's application.
5. Click the **Appmarks** tab, then review the default settings for the appmark.

   No configuration is needed. Public access is enabled automatically. If you disable public access, the appmark does not appear on the landing page until you map authorizations to set entitlements for user roles (groups).
6. Click **OK** to save the configuration.
7. (Optional) Repeat Step 2 to Step 6 to configure additional connectors for Basic SSO.
8. On the Admin page, click **Apply** to commit the changes to the appliance.
9. Wait until the configuration changes have been applied on each node of the CloudAccess cluster.
10. (Conditional) If public access is disabled, perform policy mapping to specify entitlements for identity source roles (groups).

    For more information, see "Mapping Authorizations" in the *CloudAccess Installation and Configuration Guide*.
11. After you complete the configuration, users can log in through CloudAccess to access the application. The CloudAccess login page URL is:

    ```
    https://appliance_dns_name
    ```

8.4 Troubleshooting Basic Single Sign-On

Use the following information to help troubleshoot issues with Basic SSO:

- Basic SSO can work with only one instance of CloudAccess. If you have two instances of CloudAccess and the user has an account for both systems, they will have issues when they try to log in to Basic SSO applications. Basic SSO uses sessions for saving and replaying users' credentials. Having multiple sessions open in the same browser will cause problems.
- The Basic SSO plugin for Internet Explorer 11 does not detect if you have a prior version of the plugin installed. Ensure that you do not already have the Basic SSO plugin installed before installing the plugin. You must uninstall a previous version of the plugin from Windows Control Panel, not from the browser Manage Add-ons window. Whereas previous versions of the plugin were named Basic SSO, the current version of the plugin is named Single Sign-On Assistant. Currently, there is no upgrade path from prior versions of the plugin.
- The Basic SSO plugin for Internet Explorer 11 does not support authentication to multiple instances of the browser. A cookie mismatch error occurs, followed by a forced logout.

9 Connector for Google Apps

The connector for Google Apps provides automated provisioning of accounts from the identity sources to Google Apps. The connector also provides federated single sign-on access to Google Apps through CloudAccess. The connector allows CloudAccess to authenticate a user against your identity sources and to share this authentication with Google Apps to establish the user's session.

CloudAccess includes this connector with the appliance. The connector appears automatically on the **Applications** palette of the Admin page.

Use the information in the following sections to configure a connector for Google Apps:

- Section 9.1, "Connector Requirements," on page 63
- Section 9.2, "Understanding Google Apps Provisioning," on page 64
- Section 9.3, "Creating a Google Service Account," on page 64
- Section 9.4, "Configuring the Connector for Google Apps," on page 65
- Section 9.5, "Enabling Google Mail Proxy," on page 66
- Section 9.6, "Configuring Multiple Connectors for Google Apps," on page 67

9.1 Connector Requirements

The connector for Google Apps creates a federated connection between CloudAccess and Google Apps. The connector uses OAuth 2.0 to create the federation.

NOTE: Since Google Apps is a SAML endpoint that does not support Single Logout, it is possible for users that were previously signed in through OTP to remain logged in to Google even though they have logged out of CloudAccess. To work around this issue, we recommend that users sign out of Google Apps using the URL https://accounts.google.com/logout (https://accounts.google.com/logout).

The connector for Google Apps supports account provisioning only for users in Active Directory, eDirectory, and JDBC identity sources. For more information, see Section 2.4.1, "Requirements for Provisioning," on page 18.

Verify that you meet the following requirements before you configure a connector for Google Apps:

- ❒ An understanding of identity federation using the OAuth 2.0 protocol
- ❒ A CloudAccess system, installed and configured
- ❒ A valid Google Apps for Business account
- ❒ An administrative account
- ❒ Minimum attributes populated in the identity source

  For more information, see "Configuring LDAP and JDBC Identity Sources" in the *CloudAccess Installation and Configuration Guide*.

9.2 Understanding Google Apps Provisioning

Using the Google Apps Admin console, you can configure your Google domain with an organizational structure. Using the same console, you can also assign or revoke Google Apps services such as Mail, Calendar, or Drive to or from specific organizational units within that organizational structure. As a result, user access to Google Apps services is controlled based on the user's location within the organizational structure.

CloudAccess provides support for provisioning users to specific organizational units previously configured in the Google Apps domain. After you have configured the Google Apps organizational structure and services using the Google Apps Admin console, you can configure CloudAccess to provision users to specific locations within that organizational structure.

By default, the connector for Google Apps places newly provisioned users into the top-level organization of your Google Apps domain. For example, if your Google Apps domain is mygmail.com, the connector places users in the mygmail.com organization. If you want all newly provisioned users to be placed in a sub-organization that you have created in your Google Apps domain, you can specify this organizational unit as the default when you configure the connector.

Instead of a default organizational unit, users can be provisioned to a specific organizational unit based on mappings you create on the CloudAccess Policy page. On the Policy page, the Google Apps organizational units are shown as User Placement type Authorizations. Mapping a User Placement overrides any default organizational unit you specify in the connector configuration.

9.3 Creating a Google Service Account

The connector for Google Apps uses an OAuth 2.0 federated connection to provision the user accounts from the identity source to Google Apps. To create this federated connection, you must create a Google service account. For more information about this process, see "Using OAuth 2.0 for Server to Server Applications" (https://developers.google.com/accounts/docs/OAuth2ServiceAccount).

You must create this service account before you can configure the connector for Google Apps. To create the service account, follow the instructions on the Google developer site: "Creating a service account" (https://developers.google.com/accounts/docs/OAuth2ServiceAccount#creatinganaccount).

When you create the service account, enable the following API:

- Admin SDK

In addition, when you create the service account, record the following information. You will need this information when you configure the connector for Google Apps:

- Service account's email address
- Path to the P12 private key file

Finally, you must delegate domain-wide authority to the service account. This step grants the service account access to the Google resources that CloudAccess needs to access to provision the user accounts and allow single sign-on. To grant domain-wide authority, follow the instructions on the Google developer site: "Delegating domain-wide authority to the service account" (https://developers.google.com/accounts/docs/OAuth2ServiceAccount#delegatingauthority).

Add the following Google scopes in a comma-separated format to grant the correct authorizations:

- `https://www.googleapis.com/auth/admin.directory.group`
- `https://www.googleapis.com/auth/admin.directory.group.member`

- `https://www.googleapis.com/auth/admin.directory.orgunit`
- `https://www.googleapis.com/auth/admin.directory.user`

NOTE: The format of the Google scopes is a comma-delimited list without any hard returns or line breaks. Ensure that when you copy these Google scopes, there are no hard returns or line breaks in the text.

9.4 Configuring the Connector for Google Apps

The connector for Google Apps provides user account provisioning and single sign-on access to Google Apps domains. After users log in to CloudAccess, OAuth authentication is used to automatically authenticate (single sign-on) users to Google Apps. Each cluster can support multiple instances of the connector.

To configure the connector:

1. (Conditional) If you want to enable user access to specific Google applications, complete the following steps:

   1a In the Google Apps Admin console, create one or more organizational structures underneath the top level container of your Google Apps domain.

   1b Click the **Google Apps** > **Services** menu option and enable or disable each available application for each selected organization.

2. Log in as an administrator to the CloudAccess administration console:

   `https://`*`appliance_dns_name`*`/appliance/index.html`

3. Drag the connector for Google Apps from the **Applications** palette to the **Applications** panel.
4. Provide a unique display name for the connector to appear on the Admin and landing pages, and also provide the administrator user name, domain, and service account email for the Google Apps for Business account.
5. (Conditional) Select the **Automatically configure SSO settings** option if you want CloudAccess to configure the single sign-on parameters for your Google domain. Otherwise, you must manually configure the parameters at Google Apps.

   or

   Click **Federation Instructions**. Read and follow the instructions provided to configure the connector for Google Apps to allow single sign-on for users.

6. (Conditional) If you want to specify a default organizational unit for newly provisioned users, expand **Advanced Options** and enter the path to the organization in the **Default OrgUnit** field.

   NOTE: You can specify a sub-organization at any level in your Google Apps organizational structure, using forward slashes, as long as you have set up that structure. For example, `mygoogle.com/employees/fulltime/salary`. If you leave this field blank, the connector places newly provisioned users into the top-level organization of the Google Apps domain.

7. (Optional) Expand **Advanced Naming Options** and specify the naming policy rule that the connector should use for each user account when provisioning users to Google Apps. For more information, see Section 2.4, “How CloudAccess Provisions User Accounts,” on page 18.
8. Click **OK**.
9. On the Admin page, click **Apply** to commit the changes to the appliance.

When the configuration changes have been applied on each node of the CloudAccess cluster, the application is available to users.

10. (Optional) If you want to provide users with access to Google Apps Mail from supported mobile devices, follow the procedure in Section 9.5, "Enabling Google Mail Proxy," on page 66.
11. (Optional) Modify default appmarks or configure new appmarks to specify how users should access the Google Apps applications. By default, the connector includes three appmarks that are configured for the Calendar, Mail, and Drive applications.

    For more information, see Chapter 5, "Configuring Appmarks for Connectors," on page 45.
12. Add users to the appropriate identity source group to trigger user account provisioning to Google Apps.
13. Perform policy mapping to specify entitlements for identity source groups.

    For more information, see "Mapping Authorizations" in the *CloudAccess Installation and Configuration Guide*.
14. (Conditional) If required, grant approvals for mapped authorizations.

User accounts that have been provisioned to Google Apps using CloudAccess must authenticate through CloudAccess. Direct logins to Google Apps are not allowed.

9.5 Enabling Google Mail Proxy

For users to have access to Google Apps Mail from supported mobile devices, you must first enable the option in CloudAccess and then instruct your users to perform some additional configuration steps in Google Apps.

NOTE: There is currently no way for Google administrators to set the IMAP and security settings for the whole domain. The settings are on an individual account basis, so end users must perform the steps indicated in the following procedure for their own accounts to enable mail proxy for mobile devices.

To enable email proxy:

1. Configure the connector for Google Apps in CloudAccess. For more information, see Section 9.4, "Configuring the Connector for Google Apps," on page 65.
2. Click the configured connector for Google Apps on the **Applications** panel, then click **Enable email proxy** and click **Apply**.
3. Instruct your end users to perform the following steps for their Google Apps account:

    3a Log in to your Google Apps account through CloudAccess to enable the account, and accept the terms and conditions.

    3b Click the **Settings** (gear) icon in the top right corner.

    3c Click the **Forwarding and POP/IMAP** tab, then select **Enable IMAP**.

    3d Click **Save Changes**.

    3e Click the **Accounts** tab, then click **Google account settings**. The Sign-in & security options open in a new browser window.

    3f In the navigation pane under **Sign-in & security**, click **Connected apps & sites**.

    3g Switch the **Allow less secure apps** option to `ON`.

9.6 Configuring Multiple Connectors for Google Apps

CloudAccess can support Google Apps domains by using multiple instances of the connector for Google Apps. Each connector instance must be configured with the unique credentials and domain information of the Google Apps domain that it serves.

NOTE: The **Enable email proxy** option is global across all instances of the Google Apps connector. (The option is either enabled or disabled for all instances.)

10 Connector for Microsoft Office 365

The connector for Microsoft Office 365 provides automated provisioning of accounts from the identity sources to Office 365. The connector also provides federated single sign-on access to Office 365 through CloudAccess. The connector allows CloudAccess to authenticate a user against your identity sources and to share this authentication with Office 365 to establish the user’s session. The connector supports the SAML 2.0 protocol or the WS-Federation protocol for federated SSO.

With WS-Federation, the connector supports federated single sign-on natively from a Microsoft Lync client or a Lync mobile app for iOS and Android devices. The user must install and configure the MobileAccess app and the Lync app to allow this interaction on a mobile device. The user signs in on the Lync login page as usual. The redirection to CloudAccess for authentication and service access is transparent for the user.

CloudAccess includes this connector with the appliance. The connector appears automatically in the **Applications** palette on the Admin page. However, you cannot configure the connector in CloudAccess until you have run the connector for Office 365 installer on your Windows Management Server to connect the web application to CloudAccess.

Each cluster supports multiple instances of the connector for Office 365, but each connector must serve a unique domain.

- Section 10.1, “How the Connector for Office 365 Works,” on page 69
- Section 10.2, “Connector Requirements,” on page 72
- Section 10.3, “Installing the Connector for Office 365,” on page 73
- Section 10.4, “Configuring the Connector for Office 365,” on page 75
- Section 10.5, “Validating the Connector for Office 365 Installation,” on page 75
- Section 10.6, “Configuring Appmarks for Office 365 Applications,” on page 76
- Section 10.7, “Changing the Configuration of the Connector,” on page 76
- Section 10.8, “Changing the Name of an Office 365 Security Group,” on page 76
- Section 10.9, “Upgrading the Connector from SAML 2.0 to WS-Federation,” on page 76
- Section 10.10, “Uninstalling the Connector for Office 365,” on page 77
- Section 10.11, “Installing Multiple Connectors for Office 365,” on page 77

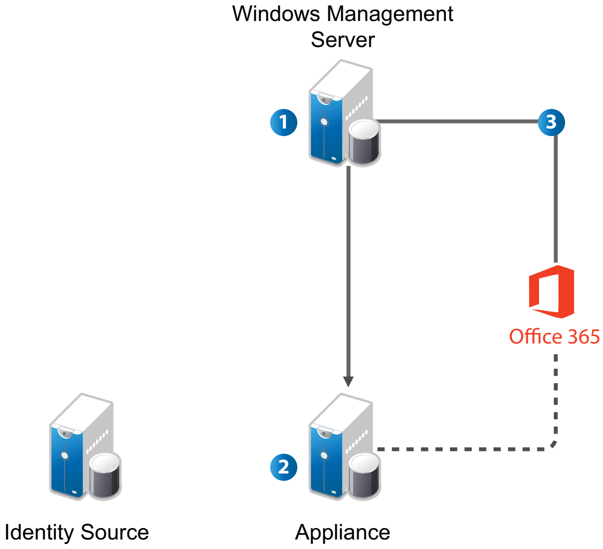
10.1 How the Connector for Office 365 Works

Before you install the connector for Office 365, review the following illustrations to help you understand how the connector works with CloudAccess.

- Section 10.1.1, “Setup and Configuration,” on page 70
- Section 10.1.2, “User Provisioning,” on page 71
- Section 10.1.3, “User Login to Office 365,” on page 71

10.1.1 Setup and Configuration

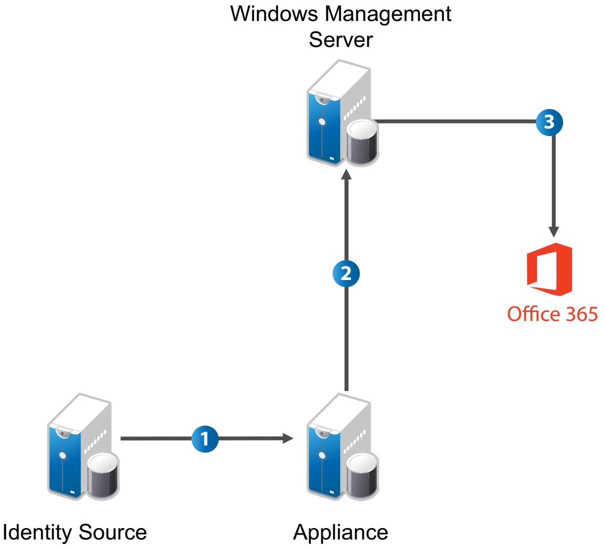
The following figure illustrates the basic setup and configuration steps.



1. Run the installer for the connector for Office 365 using the `.msi` file. For more information, see Section 10.3, "Installing the Connector for Office 365," on page 73.
2. Create a trust relationship between the CloudAccess appliance and Office 365.
3. Verify that the required attributes are populated in the identity source. For more information, see "Configuring LDAP and JDBC Identity Sources" in the *CloudAccess Installation and Configuration Guide*.
4. Configure the connector and map entitlements to Office 365. For more information, see Section 10.4, "Configuring the Connector for Office 365," on page 75.

10.1.2 User Provisioning

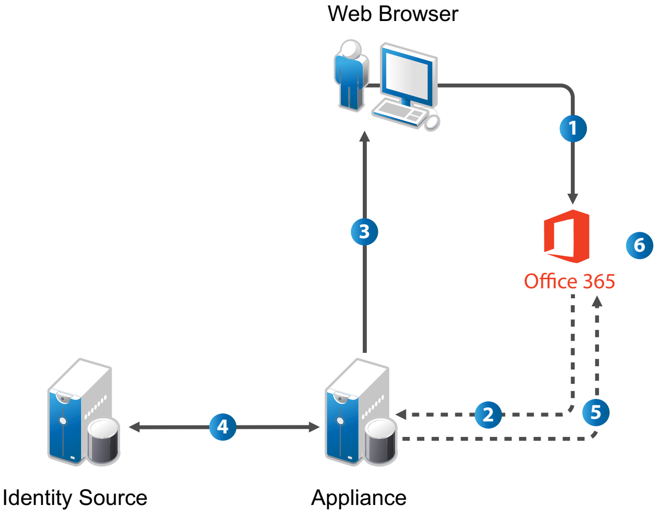
The following figure illustrates the workflow in provisioning users.



1. The administrator defines a policy to authorize access to Office 365 applications.
2. CloudAccess detects new and updated user information from the identity source.
3. CloudAccess sends user creation, license assignment, update, or deletion requests to the Windows Management Server.
4. The Windows Management Server forwards requests to Office 365 using the Windows Azure Active Directory Module for Windows PowerShell cmdlets.

10.1.3 User Login to Office 365

The following figure illustrates the workflow for users logging in to Office 365 applications.



1. The user attempts to log in to Office 365.
2. The login is redirected to CloudAccess.

3. CloudAccess prompts the user for the user name and password. Or, if Kerberos is configured, CloudAccess performs seamless authentication.
4. CloudAccess verifies the user name and password using the identity source. Or, if Kerberos is configured, CloudAccess validates the Kerberos token.
5. CloudAccess provides an assertion to Office 365.
6. Office 365 validates the assertion and allows the user access to assigned Office 365 applications.

10.2 Connector Requirements

The connector for Office 365 supports account provisioning for users only in Active Directory, eDirectory, and JDBC identity sources. For more information, see Section 2.4.1, “Requirements for Provisioning,” on page 18.

Complete the following steps before you install the connector for Office 365:

□ Identify an existing Office 365 administrative account to use, or create a new administrative account. This administrative user must not belong to the Office 365 domain that CloudAccess will manage.

□ Microsoft does not support subdomains having different federated settings than their parent. To use a subdomain for Office 365, ensure that either you do not use Office 365 with the parent domain, or that both the parent domain and its subdomain have the identical federation settings.

□ Identify the verified Office 365 domain for which CloudAccess will manage authentication.

□ Identify a Windows Management Server on which to install the connector. The Windows Management Server that you use for the connector for Office 365 should be a dedicated server that is not used for other web applications. However, the connector does not need to be installed on a domain controller or even need to be part of the domain where the CloudAccess appliance is installed. The Windows server can be a standalone server, as long as it meets the following requirements:

- Windows Server 2012 R2 or Windows Server 2008 R2 operating system with all available updates installed.
- Microsoft IIS. Install the IIS component that comes with the Windows Server operating system on your Windows Management Server. Install the Web Server (IIS) role and verify that the Application Development ASP and .Net features are installed. Enable HTTPS on IIS. The connector uses HTTPS between the CloudAccess appliance and the Office 365 web application in IIS.
- Microsoft .NET Framework 4.x. You can download .NET from the .NET downloads (http://www.microsoft.com/en-us/download/search.aspx?q=.net%20framework) web page.
- Microsoft Online Services Sign-In Assistant 7.x. You can download the Microsoft Online Services Sign-In Assistant software from the following location: Microsoft Online Services Sign-In Assistant for IT Professionals BETA (http://www.microsoft.com/en-us/download/details.aspx?id=39267). Select the `msoidcli_64.msi` file.
- Windows Azure AD Module for Windows Powershell. You can download the module from the following location: Manage Windows Azure AD using Windows PowerShell (http://technet.microsoft.com/en-us/library/jj151815.aspx#bkmk_installmodule).
- Port 443 must be open. This port is used for inbound provisioning information from CloudAccess, and outbound for Office 365 configuration and provisioning.

❒ (Conditional) If you do install the connector for Office 365 on a domain-joined server, you must be logged in as a domain administrator rather than a local machine administrator when running the installer.

❒ The Microsoft Lync support is available only if you configure the connector with WS-Federation.

❒ (Conditional) If you plan to use the Enhanced Client Profile (ECP), also called *HTTP proxy authentication*, in Microsoft Outlook, or if you plan to use Microsoft Lync, ensure that you configure CloudAccess with the following:

- A publicly resolvable, publicly accessible IP address. You can use port forwarding to protect your appliance behind your corporate firewall.

  When the user logs in to the Office 365 online portal, the browser handles all of the redirects and name resolution, so you can manually edit entries in the device's `../etc/hosts` files to work around name resolution. However, with ECP and Lync, Office 365 actually sends an authentication request directly to CloudAccess, so its IP address must be publicly accessible.
- An SSL certificate signed by a trusted certificate authority (CA) such as Verisign, Thawte, Symantec, Digicert, and so on. The certificate common name must match the appliance hostname.

NOTE: CloudAccess also supports ECP for Microsoft Exchange email on mobile devices running Android or iOS. Users must add an Exchange account on their device and enter their Exchange credentials. For more information, see the following Microsoft web pages:

- *Set up email on an Android phone or tablet*
- *Set up email on Apple iPhone, iPad, and iPod Touch*

If you use WS-Federation for the connector for Office 365, CloudAccess also supports ECP for Microsoft Lync. Users must add a Lync account on their mobile device and enter their Lync credentials. For more information, see the following Microsoft web pages:

- *Getting started with Lync 2013 for Android*
- *Getting started with Lync 2013 for iPhone*
- *Getting started with Lync 2013 for iPad*

❒ Verify that you have the minimum attributes populated in the identity source.

For more information, see "Configuring LDAP and JDBC Identity Sources" in the *CloudAccess Installation and Configuration Guide*.

10.3 Installing the Connector for Office 365

You must install the connector for Office 365 on a Windows Management Server. Before you install the connector, ensure that your environment meets all requirements stated in Section 10.2, "Connector Requirements," on page 72.

NOTE: The connector for Office 365 that is available in the Application Connector Catalog is not designed to be used with CloudAccess. That connector supports only SAML 2.0 and is very limited in functionality. Instead, ensure that you obtain the full WS-Federation connector that works with CloudAccess from the Downloads site (https://dl.netiq.com).

To configure the server and install the connector:

1 Obtain the credentials for an Office 365 administrative account. For more information, see the Office 365 website.

2 Add the federated domain name to Office 365 that will be used for single sign-on with CloudAccess and Office 365, and then validate the ownership. Use the instructions at the following web page: Add your users and domain to Office 365 (https://support.office.com/en-us/article/Add-your-users-and-domain-to-Office-365-6383f56d-3d09-4dcb-9b41-b5f5a5efd611).

> **NOTE:** Microsoft requires that each Office 365 federated domain be configured with a unique issuer ID. Thus, each instance of the connector for Office 365 connects to only one unique Office 365 federated domain.

3 Verify that the Windows server where you plan to install the connector meets the prerequisites. For more information, see Section 10.2, “Connector Requirements,” on page 72.

4 As an administrator on the Windows server, perform the following steps. For more information, see the IIS Manager help.

4a Install the Web Server (IIS) role and verify that the Application Development ASP and .Net features are installed.

4b Create a self-signed certificate in IIS Manager. Alternatively, you can use an imported server certificate. For more information, see Importing a Server Certificate (http://technet.microsoft.com/en-us/library/cc732785%28v=ws.10%29.aspx).

4c Add an HTTPS binding for the Default Web Site using the certificate you created.

4d Test HTTPS by accessing the server through a browser using HTTPS. You should see the IIS server page after a certificate error (if you are using a self-signed certificate).

4e In the Application Pool Settings in IIS Manager, verify that the DefaultAppPool version is .NET v4.0.

4f Restart the IIS service.

4g Install Microsoft Online Services Sign-In Assistant 7.x. For more information, see Section 10.2, “Connector Requirements,” on page 72.

4h Install Windows Azure AD Module for Windows. For more information, see Section 10.2, “Connector Requirements,” on page 72.

5 As an administrator on the Windows server, download the connector for Office 365 `.zip` file from NetIQ Downloads (https://dl.netiq.com/). Unzip the file and run the Windows `netiq-office365-connector-`*`x.x.x`*`.msi` installer. You will need the following information:

- DNS name of the CloudAccess appliance.
- Administrator name and password of the CloudAccess appliance.
- User name and password for the Office 365 Global administrator account.
- The federated domain name specified in Step 2. If you get an error during installation, ensure that you selected the correct domain name.

Alternatively, you can run the connector installer in “silent mode” from the command line as follows:

```
msiexec /i netiq-office365-connector-x.x.x.msi /qb /L*v install-log.txt
AG4CHOSTNAME="Appliance_Admin_DNS" AG4CADMIN="Appliance_Admin_Name"
AG4CADMINPASS="Appliance_Admin_Password" O365ADMIN="O365_Admin"
O365ADMINPASS="O365_Password" O365FEDDOMAIN="Domain_DNS"
O365USAGELOCATION="US" LOCALIP="Windows_Server_IP_Address"
APPPOOLNAME_NT=".NET v4.5"
```

By default, CloudAccess does not generate an installation log when you install the connector for Office 365. If you want a log of the installation, you must launch the installer from the command line using the following command:

```
msiexec /i netiq-office365-connector-x.x.x.msi /L*V "C:\log\example.log"
```

IMPORTANT: The connector for Office 365 installation location is `c:\NetIQ\Office365Connector`. You cannot change this location.

6 Continue with configuration of the connector in CloudAccess. For more information, see Section 10.4, “Configuring the Connector for Office 365,” on page 75.

10.4 Configuring the Connector for Office 365

After you have installed the connector, no specific configuration of the connector itself is required. However, you should review available naming policy options and create additional appmarks for Office 365 applications before you provision users.

To configure the connector:

1 Log in as an administrator to the CloudAccess administration console:

```
https://appliance_dns_name/appliance/index.html
```

2 Click the connector for Office 365 on the **Applications** panel, then click **Configure**.

3 On the **Configuration** tab:

  3a Expand **Advanced Options** to access naming policy options.

  3b Review naming policy options and, if appropriate for your organization, specify the rule you want the connector to follow when provisioning users to Office 365. For more information about naming policies, see Section 2.4.6, “Understanding Naming Policy Options,” on page 24.

4 Click the **Appmarks** tab and configure appmarks for different Office 365 applications. For more information, see Section 10.6, “Configuring Appmarks for Office 365 Applications,” on page 76.

5 Click **OK**, then click **Apply**.

6 Continue with policy mapping to set user entitlements to the appropriate applications and provision user accounts. For more information, see “Mapping Authorizations” in the *CloudAccess Installation and Configuration Guide*.

10.5 Validating the Connector for Office 365 Installation

After you have installed the connector, perform the following steps to validate the installation:

1 Verify that the Policy Mapping page displays Identity Source Groups on the left side and the connector for Office 365 on the right side.

2 Map one of your groups to the User Authorizations, then verify that CloudAccess provisioned your users.

3 Log in to Office 365 at http://www.office365.com as a provisioned user.

4 Specify the user name of `user@domain` where the domain is the federated domain name specified in Step 2 on page 74.

NOTE: If you have any issues with the connector, check the Windows Event Viewer on the Windows server where the connector is installed. You can view all events for the connector to help troubleshoot those issues. In the Windows Event Viewer, expand **Windows Logs**, then click **Application**.

10.6 Configuring Appmarks for Office 365 Applications

By default, the connector for Office 365 includes a single appmark that is configured for the user's home page. You can modify this default appmark or create additional appmarks as needed. Appmarks that are then mapped in Policy Mapping appear on the landing page or in the MobileAccess app for users to whom you have granted access. For more information about configuring appmarks, see Chapter 5, "Configuring Appmarks for Connectors," on page 45.

NOTE: If you configure appmarks for users to launch Office 365 applications using Safari on mobile devices, you should instruct users to set Safari to never block cookies. Alternatively, consider selecting another **Launch with** option to ensure that users do not experience logout errors. For more information, see "Office 365 Logout Error on Mobile Devices".

10.7 Changing the Configuration of the Connector

If you change the federated domain name, you must reinstall the connector for Office 365. The installation changes the configuration information in the connector. Delete the existing connector, then run through the installation again using the new federated domain name.

10.8 Changing the Name of an Office 365 Security Group

CloudAccess tracks the security groups in Office 365 by their group name. If you change the name of a security group in Office 365, the rename appears as a delete and add to CloudAccess. CloudAccess deletes the old group and adds the new group. Any authorizations that are mapped to the deleted group are also removed. After you rename a security group in Office 365, verify that the group appears in CloudAccess with its new name, then go to the Policy page and remap the authorizations for Office 365 to the group.

10.9 Upgrading the Connector from SAML 2.0 to WS-Federation

In CloudAccess 2.1 or later, the connector uses WS-Federation to support single sign-on from a Microsoft Lync client. In previous releases, CloudAccess uses SAML 2.0 for single sign-on and provisioning. You must upgrade an existing connector for Office 365 to take advantage of the Lync access capability.

To upgrade the connector for Office 365:

1. Log in to the administration console.
2. On the Admin page in the Applications panel, click the connector for Office 365, then select **Configure**.
3. In the Configuration window, select **Use WS-Federation**.

4. Select **Automatically Configure SSO Settings** to let the connector modify the federation settings from SAML to WS-Federation.

   If you do not select this option, you must manually reconfigure the **Federation Settings**.

5. Click **OK**.
6. On the Admin page, click **Apply** to commit the changes to the appliance.
7. Wait until the configuration changes have been applied on each node of the CloudAccess cluster.

10.10 Uninstalling the Connector for Office 365

The connector for Office 365 consists of multiple components. To correctly uninstall the connector, either follow the steps below to use the uninstall function in Windows Control Panel or run the following command on the Windows server:

```
msiexec /x netiq-office365-connector-x.x.x.msi /qb
```

IMPORTANT: Do not manually remove the connector for Office 365 from the CloudAccess administration console. If you just delete the connector for Office 365 icon from the Admin page, all of the components on the Windows server still exist and run. This causes issues in CloudAccess if you need to reinstall the connector, unless you run the connector uninstall from the Windows server before attempting to reinstall a new connector.

To uninstall the connector for Office 365:

1. Verify that the CloudAccess appliance is running.

   Once the connector is uninstalled from the Windows server, it is also automatically removed from the CloudAccess appliance, so the appliance must be running.

2. Log in to the Windows server as an administrator.
3. Uninstall the connector for Office 365 using Windows Control Panel.
4. Log in to the CloudAccess administration console using an appliance administrator account and verify that the connector has been removed from the console.

10.11 Installing Multiple Connectors for Office 365

CloudAccess supports multiple connectors for Office 365. However, each connector must connect to a unique Office 365 domain, and you must install each connector on a separate Windows server.

11 Connector for Salesforce

The connector for Salesforce provides automated provisioning of user accounts from the identity sources to Salesforce. The connector also provides federated single sign-on access to Salesforce with SAML 2.0 through CloudAccess. The connector allows CloudAccess to authenticate a user against your identity sources and to share this authentication with Salesforce to establish the user's session.

In addition, the connector supports single sign-on natively to a Salesforce mobile app for iOS and Android devices. The user must install and configure the MobileAccess app and the Salesforce app to allow this interaction. The user signs in on the Salesforce login page as usual.

CloudAccess includes this connector for Salesforce with the appliance. The connector is located on the **Applications** palette of the Admin page.

Use the information in the following sections to configure a connector for Salesforce:

- Section 11.1, "Connector Requirements," on page 79
- Section 11.2, "Configuring Salesforce to Trust CloudAccess," on page 80
- Section 11.3, "Configuring the Connector for Salesforce," on page 80
- Section 11.4, "Configuring Appmarks for Salesforce," on page 82
- Section 11.5, "Configuring Multiple Connectors for Salesforce," on page 83
- Section 11.6, "Using SSO to Salesforce on Mobile Devices," on page 83
- Section 11.7, "Configuring Delegated Authentication in Salesforce," on page 86
- Section 11.8, "Configuring the Salesforce Federation Identifier," on page 87

11.1 Connector Requirements

The connector for Salesforce supports account provisioning only for users in Active Directory, eDirectory, and JDBC identity sources. For more information, see Section 2.4.1, "Requirements for Provisioning," on page 18.

Verify that you meet the following requirements before you configure the connector for Salesforce:

- ❒ An understanding of identity federation using the SAML 2.0 protocol.

  For more information about SAML, see the OASIS website (https://wiki.oasis-open.org/security/FrontPage).
- ❒ A full or developer Salesforce account with provisioning APIs enabled.
- ❒ Administrator access to the Salesforce account. An understanding of Salesforce and its account management tools are presumed.
- ❒ (Conditional) A security token from Salesforce.

  For more information, see Section 11.2, "Configuring Salesforce to Trust CloudAccess," on page 80.
- ❒ The location in the Salesforce administration console where you will configure the SAML 2.0 federation for CloudAccess.

When you configure the connector, the **Federation Instructions** provide the information that you will need to set up the federation in Salesforce for CloudAccess. This information includes the metadata specific to the appliance; a signing certificate for the appliance; the field values to use; and other guidance.

❒ The metadata file from Salesforce.

You generate and download this file after you configure SAML 2.0 federation for CloudAccess in Salesforce.

❒ Minimum attributes populated in the identity source.

For more information, see "Configuring LDAP and JDBC Identity Sources" in the *CloudAccess Installation and Configuration Guide*.

11.2 Configuring Salesforce to Trust CloudAccess

We recommend that you configure Salesforce to trust the IP address of the CloudAccess appliance.

To add the CloudAccess IP address as a trusted source to Salesforce:

1. Log in to the Salesforce Admin tools web page.
2. Click **Administration Setup** > **Security Controls** > **Network Access**.
3. Specify the IP address of the CloudAccess appliance.

   or

   If you are in a clustered environment, specify the IP address of the L4 switch.

If you do not configure Salesforce to trust the IP address of the CloudAccess appliance, you must obtain a security token from Salesforce and append the security token to the administrator password specified when you configure the connector for Salesforce.

For example, if your Salesforce administrator account password is `Test1234` and the Salesforce security token is `XyZ`, the **Password** field must contain `Test1234XyZ`.

11.3 Configuring the Connector for Salesforce

The phone icon that CloudAccess displays on each configured instance of the connector for Salesforce indicates that Delegated Authentication can be used with Salesforce. All of the configuration required for using Delegated Authentication with the appliance is done at Salesforce. For more information, see Section 11.7, "Configuring Delegated Authentication in Salesforce," on page 86.

You must go back and forth between the CloudAccess Admin page and the Salesforce administration page to configure the connector.

To configure the connector for Salesforce:

1. Do one of the following:
   - Configure Salesforce to trust CloudAccess.
   - Obtain a security token from Salesforce.

   For more information, see Section 11.2, "Configuring Salesforce to Trust CloudAccess," on page 80.
2. (Optional) Log in to Salesforce as the account administrator, then enable and configure Salesforce Delegated Authentication single sign-on for your Salesforce organization.

For more information, see Section 11.7, “Configuring Delegated Authentication in Salesforce,” on page 86.

3. Log in with an appliance administrator account to the CloudAccess administration console at

   ```
   https://appliance_dns_name/appliance/index.html
   ```

4. Drag the connector for Salesforce from the **Applications** palette to the **Applications** panel.
5. Specify a unique display name for the connector to appear on the Admin page.
6. Specify the login credentials for the Salesforce administrator user.

   NOTE: If you opted not to have CloudAccess as a trusted source for Salesforce in Step 1, you must append the security token to the Salesforce administrator’s password.

7. In the **Environment** field, specify whether you have a Production, Development, or Sandbox Salesforce environment. The login URL that is used to verify your Salesforce credentials can be different for each of these environments.
8. Select or deselect **Delegated Authentication single sign-on is disabled in Salesforce**, according to your action for Step 2.
9. (Conditional) If delegated authentication is disabled and if you want to give users control of when their accounts are provisioned, select **Prompt users for an existing Salesforce account before provisioning**.

   For more information about account provisioning, see Section 2.4, “How CloudAccess Provisions User Accounts,” on page 18.

10. Click **Advanced Settings**, and then specify whether the **Federation attribute** should use a GUID or the user’s network identity retrieved from the identity source. The Federation attribute stores the user’s Salesforce federation ID.

    For more information, see Section 11.8, “Configuring the Salesforce Federation Identifier,” on page 87.

11. Expand the **Federation Instructions**, then copy and paste the instructions into a text file to use during the Salesforce configuration for single sign-on.

    NOTE: You must use a text editor that does not introduce hard returns or additional white space. For example, use Notepad instead of Wordpad.

12. Click **OK** to save the configuration so far while you configure Salesforce to work with CloudAccess.

    The configuration for the connector for Salesforce is not yet complete.

13. Log in to Salesforce as the account administrator, then configure the SAML 2.0 federation for CloudAccess in the Salesforce administration console.

    Use the information from the **Federation Instructions** in Step 11 to complete the setup.

    NOTE: When you copy the appliance’s signing certificate, ensure that you include all leading and trailing hyphens in the certificate’s Begin and End tags.

14. After you configure federation for CloudAccess in Salesforce, generate and download the Salesforce metadata file.
15. On the CloudAccess Admin page, click the connector for Salesforce, then click **Configure**.
16. Upload the Salesforce metadata file that you downloaded in Step 14 to the connector for Salesforce.

17 (Optional) Expand **Advanced Naming Options** and specify the naming policy and collision handling rule that the connector should use for each user account when provisioning users to Salesforce. For more information, see Section 2.4, "How CloudAccess Provisions User Accounts," on page 18.

18 Click the **Appmarks** tab, then review and edit the default settings for the appmark.

For more information, see Section 11.4, "Configuring Appmarks for Salesforce," on page 82.

19 Click **OK** to save the configuration.

20 On the Admin page, click **Apply** to commit the changes to the appliance.

21 Wait until the configuration changes have been applied on each node of the CloudAccess cluster.

22 Click **Policy** in the toolbar, then perform policy mapping to specify entitlements for identity source roles (groups).

For more information, see "Mapping Authorizations" in the *CloudAccess Installation and Configuration Guide*.

23 After you complete the configuration, users can log in through CloudAccess to single sign-on to Salesforce. The CloudAccess login page URL is:

```
https://appliance_dns_name
```

For information about single sign-on through the Salesforce mobile app, see Section 11.6, "Using SSO to Salesforce on Mobile Devices," on page 83.

11.4 Configuring Appmarks for Salesforce

By default, the connector for Salesforce includes a single appmark that is configured for the user's landing page. You can configure the appmark for the desktop browser and supported mobile devices. You can modify this default appmark or create additional appmarks as needed.

NOTE: To enable single sign-on to the mobile Salesforce app, you can select **Native application** from the list of **Launch with** options on the **Appmarks** tab. However, this is not a requirement.

Appmarks that are then mapped in Policy Mapping appear on the landing page and, if configured, in the MobileAccess app for entitled users. For more information, see Chapter 5, "Configuring Appmarks for Connectors," on page 45.

CloudAccess automatically updates users' Salesforce profiles when their policy mapping changes. Salesforce enforces the following restrictions:

- Salesforce updates the user's profile to the new profile only if licenses are available. The connector driver log shows errors if licenses are not available.
- Salesforce will not move a user from a paid license to a free license. Salesforce also enforces this restriction in its administration console. To free a license for a paid account, you must de-activate the user.

11.5 Configuring Multiple Connectors for Salesforce

Each cluster supports multiple connectors for Salesforce. If you want to configure more than one instance of the connector for Salesforce, each Salesforce account must be configured with a unique URL. Configuring the URL requires Salesforce assistance, and it often takes at least a day to complete.

To configure the Salesforce URL:

1. Log in to the Salesforce administration web page.
2. Click **Administration Setup** > **Domain Management** > **My Domain**.
3. Provide a unique subdomain name for your organization and click **Check Availability**.
4. If the subdomain you specified is available, select the check box to indicate that you agree to the terms and conditions, then click **Register Domain**.
5. Wait for Salesforce to register your domains. This process takes time.

After the registration is complete, Salesforce provides you with a URL that supports SP-initiated logins and is similar to the following:

```
https://<custom_name>.mysalesforce.com
```

11.6 Using SSO to Salesforce on Mobile Devices

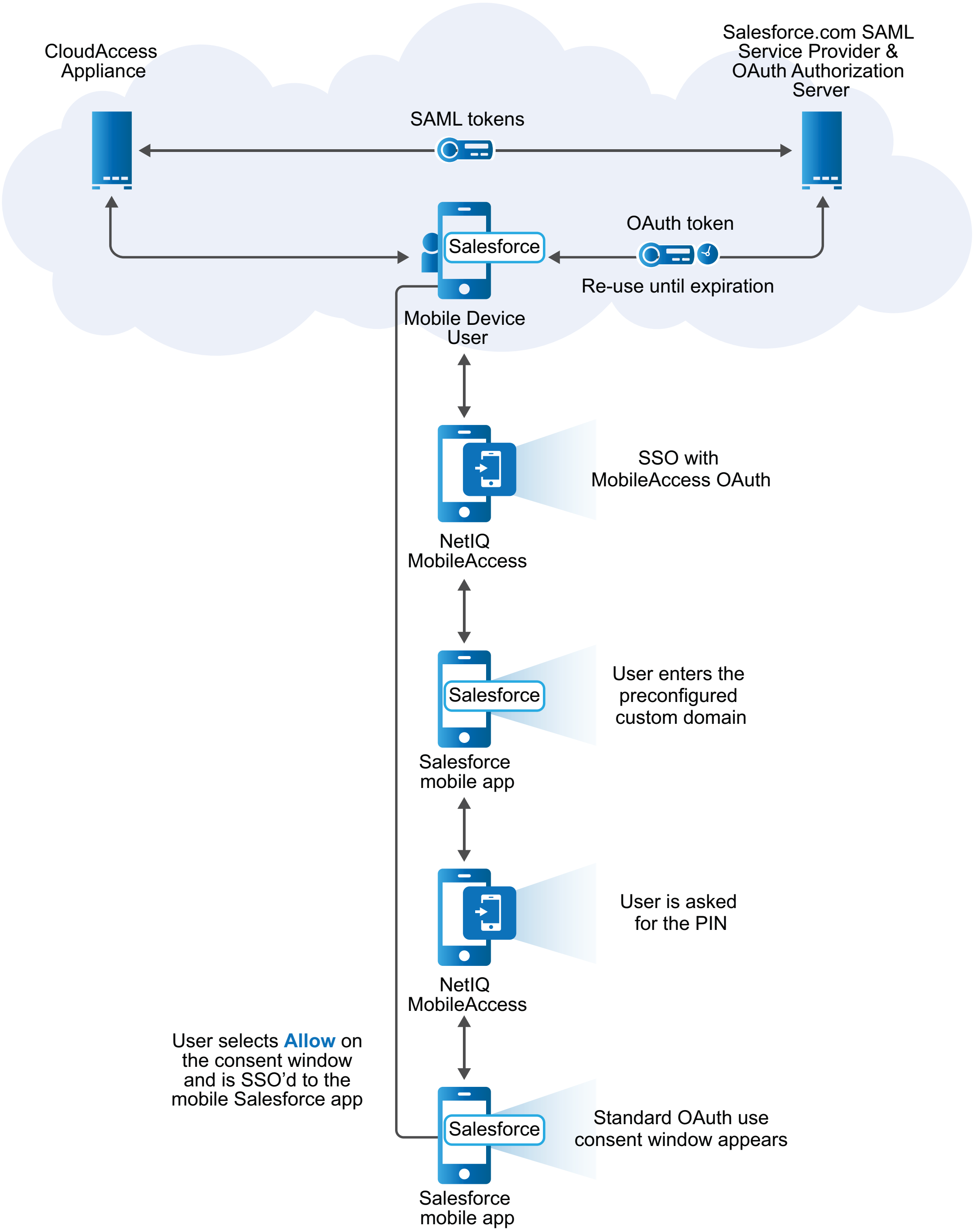
Salesforce now offers a combination of OAuth and SAML functionality to provide seamless SSO facilities not only for web browsers, but also desktop and mobile applications. By using OAuth to enable users to connect applications to their accounts, and leveraging SAML for the authentication of that connection, the single sign-on integration that was previously applicable only for the web browser can now service mobile applications.

The CloudAccess connector for Salesforce provides the necessary protocols and interfaces to the SAML and OAuth features of Salesforce. Single or multi-factor authentication can be used. The mobile device provides a standard browser interface to end users by using HTML. SSL/TLS is used for all appliance connections to protect user credentials and tokens.

11.6.1 Understanding the Mobile SSO Process

The following illustration provides a high-level overview of the mobile SSO to Salesforce process.

Figure 11-1 *SSO to Mobile Salesforce Process*


CloudAccess Appliance
Salesforce.com SAML Service Provider & OAuth Authorization Server
SAML tokens
Salesforce
OAuth token
Re-use until expiration
Mobile Device User
SSO with MobileAccess OAuth
NetIQ MobileAccess
Salesforce
User enters the preconfigured custom domain
Salesforce mobile app
User is asked for the PIN
NetIQ MobileAccess
User selects Allow on the consent window and is SSO'd to the mobile Salesforce app
Salesforce
Standard OAuth use consent window appears
Salesforce mobile app

From the CloudAccess administrator's perspective, the process is as follows:

1. The CloudAccess administrator sets up a SAML trust relationship between the CloudAccess appliance and the Salesforce service to provide user authentication and SAML tokens.
2. The end user authenticates with the CloudAccess appliance to obtain a SAML token and send it to the Salesforce.com server.

3. The Salesforce.com server accepts a valid SAML token from CloudAccess and issues an OAuth token for the mobile device.
4. The Salesforce native app stores and uses an OAuth token to access Salesforce.com services. The token is reused for future sessions, so the user does not have to re-enter credentials as long as the token has not expired.

From the user’s perspective, the process is as follows:

1. The user installs and sets up the mobile Salesforce app on a supported mobile device.
2. The user installs the MobileAccess app on a supported mobile device.
3. The user opens the Salesforce mobile app on the device and enters the preconfigured custom domain.
4. The user is redirected to the MobileAccess app and, if required, is asked for the PIN.
5. The user is redirected back to the Salesforce app and the standard OAuth use consent window appears.
6. The user selects **Allow** on the consent window and is SSO’d to the mobile Salesforce app. The user does not have to re-enter credentials until the OAuth session token expires.
7. When the user logs out of the mobile Salesforce app, the user sees the Salesforce.com login page.

NOTE: If the user does not have the MobileAccess app installed on the device (or if it was previously installed and then deleted), instead of being automatically authenticated, a MobileAccess login screen appears in step 4 and the user must enter credentials.

11.6.2 Requirements for Mobile SSO to Salesforce

The following requirements must be met to enable SSO to Salesforce on mobile devices:

❐ When you complete the connector configuration at Salesforce to allow single sign-on for users, you must configure a custom domain at Salesforce and provide the custom domain URL to users.

For more information, see Section 11.3, “Configuring the Connector for Salesforce,” on page 80.

❐ When you configure the connector for Salesforce in CloudAccess, you may create an appmark specifically for single sign-on to the mobile Salesforce app, selecting **Native application** from the list of **Launch with** options.

For more information, see Chapter 5, “Configuring Appmarks for Connectors,” on page 45.

❐ SAML SSO works on mobile devices only if the MobileAccess app is also installed and configured on the device. Without the MobileAccess app installed on mobile devices, only delegated authentication using SAML is available.

11.7 Configuring Delegated Authentication in Salesforce

Salesforce allows two different types of authentication methods: SAML and delegated authentication. By default, Salesforce activates only the SAML authentication. SAML is available for browser-based authentication or for mobile devices. However, SAML SSO works on mobile devices only if the MobileAccess app is also installed and configured on the device.

Delegated authentication must be activated on a per-Salesforce organization basis. This allows CloudAccess to support users authenticating with mobile devices as well as users authenticating with browsers.

The phone icon that CloudAccess displays on all the Salesforce connectors indicates that Delegated Authentication can be used with Salesforce. You must enable and configure Delegated Authentication in Salesforce, and enable it in the connector. For more information, see Step 2 and Step 8 in Section 11.3, "Configuring the Connector for Salesforce," on page 80.

The following setup is required in CloudAccess in order for delegated authentication to work properly:

- The DNS name of the CloudAccess cluster must be publicly resolvable.
- The SSL certificate must be signed by a well-known certificate authority (CA).

To configure Salesforce for delegated authentication:

1. Follow the instructions in the Salesforce documentation to enable delegated authentication single sign-on for your organization.

   For more information, see Configuring Salesforce for Delegated Authentication (https://login.salesforce.com/help/doc/en/sso_delauthentication_configuring.htm).

2. After delegated authentication has been enabled at Salesforce, complete the following configuration steps:

   2a Log in to the Salesforce administration page.

   2b Click **Your Name** > **Setup** > **Security Controls** > **Single Sign-On Settings** > **Edit**.

   2c In the **Delegated Gateway URL** field, specify a value similar to the following: `https://`*`cloudaccess_public_dns_name`*`/osp/a/t1/auth/external/sfda`.

   2d Do not select **Force Delegated Authentication Callout**.

   This option affects the performance of user logins.

   2e Enable the **Is Single Sign-On Enabled** permission. Note that if you want to prompt users to validate their accounts, you must disable this option instead. For more information about the **Prompt users for an existing account before provisioning** option, see Section 2.4, "How CloudAccess Provisions User Accounts," on page 18.

3. Configure a connector for Salesforce in CloudAccess as described in section Section 11.3, "Configuring the Connector for Salesforce," on page 80, but deselect the **Delegated authentication single sign-on is disabled in Salesforce** option.

When end users authenticate to Salesforce through their mobile devices, they will authenticate entering identity source credentials, where the user name is specified in email format to match the user name in the Salesforce account.

For example, if Active Directory user `Ted` with password `password` has been provisioned to Salesforce domain `mydomain-dev-ed.my.salesforce.com`, the user name for login from a mobile device app such as Salesforce Chatter would be `Ted@mydomain-dev-ed.my.salesforce.com` and the password would be `password`.

11.8 Configuring the Salesforce Federation Identifier

The Salesforce connector uses the Federation attribute to store the user's Salesforce federation identity. You can use one of the following as the attribute type for all users:

- **GUID:** The federation identifier uses the adroitBISObjectID. Using a GUID is the default setting.
- **WorkforceID/employeeID:** If you select this option, the identity source must supply the value for the appliance and Salesforce connector to use. All of the current identity sources support a workforceID attribute:
  - workforceID (eDirectory)
  - employeeID (Active Directory)
  - workforceid (JDBC)

  If the user has a workforceID or employeeID in the identity source, the user's account is provisioned in Salesforce.

  If the user does not have a workforceID or employeeID in the identity source, the `connectors_SFORCE_XXXXX.log` file has a message that the provisioning activity for that user was vetoed. Add a workforceID/employeeID to the User object in the identity source. When the workforceID or employeeID is synchronized, the account is automatically provisioned.

When you configure the Salesforce connector, you can use the **Advanced Settings** > **Federation attribute** option to specify which attribute type to use. For more information, see Section 11.3, "Configuring the Connector for Salesforce," on page 80.

To change from using a GUID to using a workforceID/employeeID for the federation identity, or vice versa:

1. Log in with an appliance administrator account to the Admin page at

   ```
   https://appliance_dns_name/appliance/index.html
   ```

2. On the Policy Mapping page, de-provision users from Salesforce by removing the current policy mapping so that the users are marked as inactive in Salesforce.

   For information about policy mapping, see "Mapping Authorizations" in the *CloudAccess Installation and Configuration Guide*.

3. On the Admin page, click the configured connector for Salesforce, then click **Configure**.
4. In the Salesforce connector configuration, click **Advanced Settings**, change the **Federation identifier** setting, then click **OK** and **Apply** to save and apply the change.
5. Redo the policy mapping to trigger re-provisioning of users to Salesforce. The federation identifier is modified to use the appropriate attribute.
6. (Conditional) If you changed the **Federal identifier** setting from GUID to workforceID/employeeID, verify that all users were provisioned.

   6a Check the `connectors_SFORCE_XXXXX.log` file for messages about any user objects that were not provisioned because they did not have a workforceID/employeeID.

   6b For each user who was not provisioned, add a workforceID/employeeID to the User object in the identity source.

   When the workforceID/employeeID is synchronized, the account is automatically provisioned.

   6c Repeat this process to ensure that all authorized users are provisioned.

12 Connector for ServiceNow

The connector for ServiceNow provides automated provisioning of user accounts from the identity sources to ServiceNow. The connector also provides federated single sign-on access to ServiceNow with SAML 2.0 through CloudAccess. The connector allows CloudAccess to authenticate a user against your identity sources and to share this authentication with ServiceNow to establish the user's session.

CloudAccess includes this connector for ServiceNow with the appliance. The connector is located on the **Applications** palette of the Admin page.

NOTE: The Application Connector Catalog also includes an SSO-only connector for ServiceNow that does not provision users. For more information, see Chapter 13, "Single Sign-On Connectors," on page 93.

Use the information in the following sections to configure a connector for ServiceNow:

- Section 12.1, "Connector Requirements," on page 89
- Section 12.2, "Configuring the Connector for ServiceNow," on page 90
- Section 12.3, "Configuring Appmarks for ServiceNow," on page 91

12.1 Connector Requirements

The connector for ServiceNow supports account provisioning only for users in Active Directory, eDirectory, and JDBC identity sources. For more information, see Section 2.4.1, "Requirements for Provisioning," on page 18.

Verify that you meet the following requirements before you configure the connector for ServiceNow:

❒ An understanding of identity federation using the SAML 2.0 protocol.

For more information about SAML, see the OASIS website (https://wiki.oasis-open.org/security/FrontPage).

❒ Administrator access to the ServiceNow account. An understanding of ServiceNow and its account management tools are presumed.

❒ The instance administrator username and password for the ServiceNow instance you want to manage.

❒ The location in the ServiceNow administration console where you will configure the SAML 2.0 federation for CloudAccess.

When you configure the connector, the **Federation Instructions** provide the information that you will need to set up the federation in ServiceNow for CloudAccess. This information includes the metadata specific to the appliance, a signing certificate for the appliance, and the field values to use.

❒ The metadata file from ServiceNow.

You generate and download this file after you configure SAML 2.0 federation for CloudAccess in ServiceNow.

❐ Minimum attributes populated in the identity source.

For more information, see “Configuring LDAP and JDBC Identity Sources” in the *CloudAccess Installation and Configuration Guide*.

12.2 Configuring the Connector for ServiceNow

You must go back and forth between the CloudAccess Admin page and the ServiceNow administration page to configure the connector for ServiceNow.

NOTE: You can configure multiple connectors, as long as each one connects to a unique instance of ServiceNow.

To configure the connector for ServiceNow:

1 Log in with an appliance administrator account to the CloudAccess administration console at

`https://`*`appliance_dns_name`*`/appliance/index.html`

2 Drag the connector for ServiceNow from the **Applications** palette to the **Applications** panel.

3 Specify a unique display name for the connector to appear on the Admin page.

4 Specify the login credentials for the administrator user of the ServiceNow instance.

5 Specify the ServiceNow URL. You can cut and paste your instance URL in the ServiceNow management portal. For example, `https://testinstance.service-now.com/`

6 (Optional) In the **Assertion** field, specify either username or email address. This field must match whatever you set in the trusted idp settings in ServiceNow when you configure the trust between CloudAccess and ServiceNow. (The choices in ServiceNow are email or user_name.)

The connector validates the credentials before saving.

7 Expand the **Federation Instructions**, then copy and paste the instructions into a text file to use during the ServiceNow configuration for single sign-on.

NOTE: You must use a text editor that does not introduce hard returns or additional white space. For example, use Notepad instead of Wordpad.

8 Click **OK** to save the configuration so far while you configure ServiceNow to work with CloudAccess.

9 Log in to ServiceNow as the account administrator.

10 Using the information from the **Federation Instructions**, configure the SAML 2.0 federation for CloudAccess in the ServiceNow management portal.

You configure trust between CloudAccess and ServiceNow by enabling single sign-on, then creating CloudAccess as a trusted identity provider for the ServiceNow instance.

NOTE: When you copy the appliance’s signing certificate, ensure that you include all leading and trailing hyphens in the certificate’s Begin and End tags.

11 After you configure federation for CloudAccess in ServiceNow, generate and download the ServiceNow metadata file.

12 On the CloudAccess Admin page, click the connector for ServiceNow, then click **Configure**.

13 Upload the ServiceNow metadata file that you downloaded in Step 11 to the connector for ServiceNow.

14 CloudAccess does not currently have any selection criteria for the naming policy for ServiceNow. Whatever the CN attribute is when the user is imported, that is the user name that is created in ServiceNow. For more information, see Section 2.4, "How CloudAccess Provisions User Accounts," on page 18.

15 Click the **Appmarks** tab, then review and edit the default settings for the appmark. For more information, see Section 12.3, "Configuring Appmarks for ServiceNow," on page 91.

16 Click **OK** to save the configuration.

17 On the Admin page, click **Apply** to commit the changes to the appliance.

Wait until the configuration changes have been applied on each node of the CloudAccess cluster.

18 Click **Policy** in the toolbar, then perform policy mapping to specify entitlements for identity source roles (groups).

For more information, see "Mapping Authorizations" in the *CloudAccess Installation and Configuration Guide*.

19 After you complete the configuration, users can log in through CloudAccess to single sign-on to ServiceNow. The CloudAccess login page URL is:

```
https://appliance_dns_name
```

12.3 Configuring Appmarks for ServiceNow

By default, the connector for ServiceNow includes a single appmark that is configured for the user's landing page. You can configure the appmark for the desktop browser and supported mobile devices. You can modify this default appmark or create additional appmarks as needed.

NOTE: The ServiceNow native mobile app is not currently supported. Mobile access to ServiceNow is available only through a browser.

Appmarks that are then mapped in Policy Mapping appear on the landing page and, if configured, in the MobileAccess app for entitled users. For more information, see Chapter 5, "Configuring Appmarks for Connectors," on page 45.

CloudAccess automatically updates users' ServiceNow user types when their policy mapping changes. ServiceNow updates the user to the new user type only if licenses are available. The connector driver log shows errors if licenses are not available.

13 Single Sign-On Connectors

The Application Connector Catalog provides a set of connectors that create a federated connection between CloudAccess and an application using SAML 2.0. The single sign-on connectors do not provision the user accounts.

Each connector has a different set of requirements. However, the steps for configuring the connector are the same regardless of the connector. Use the following information to configure your single sign-on connectors.

13.1 Global Requirements for the Single Sign-On Connectors

The following requirements are the same for all single sign-on connectors. Ensure that you meet the following requirements before configuring a single sign-on connector:

- ❐ An understanding of identity federation using the SAML 2.0 protocol.
- ❐ A user account for each user who wants access to the single sign-on service. The single sign-on connectors do not provision user accounts.
- ❐ (Optional) An X.509 signing certificate from the application is required to support single logout. Communications use SSL regardless of whether you provide this certificate.

13.2 Requirements for the Single Sign-On Connectors

Use the information in the following table to gather connector-specific requirements.

***Table 13-1** Connector Specific Requirements*

Connector	Requirements
Connector for Accellion (SAML 2.0)	❐ An enterprise Accellion account. ❐ The Accellion domain name and the application ID. The Accellion administrative URL contains both elements. `http://`*`domain_name`*`.accellion.net/courier/`*`application_id`*`/index.html` ❐ The administration link in Accellion to configure Single Sign-On (SSO).
Connector for ADFS (SAML 2.0)	❐ An ADFS 2.0 system ❐ The metadata file from the ADFS 2.0 system. `https://`*`adfsserver`*`/FederationMetadata/2007-06/FederationMetadata.xml`

Connector	Requirements
Connector for Box (SAML 2.0)	□ An enterprise level Box account. Obtaining an enterprise Box account requires you to provide configuration details for your organization so Box can help set up the SSO connection.
Connector for Jive (SAML 2.0)	□ A Jive account. **NOTE:** There are three types of Jive accounts: Cloud, Hosted, or On Prem. If you have a Hosted or On Prem account, you have access to the federation settings required to configure the connector for Jive. If you have a Cloud account, you must contact Jive technical support to configure the federation between Jive and CloudAccess. □ The base metadata URL from your Jive instance.
Connector for ServiceNow (SAML 2.0)	□ A management ServiceNow account created with the SAML 2.0 Update 1 plugin. □ The ServiceNow instance name. For example, `http://`*`your_instance`*`.service-now.com/`.
Connector for VMware vCloud (SAML 2.0)	□ A VMware vCloud deployment with a vCloud director and the following information from your vCloud director: □ DestinationURL □ vCloud Host IP address □ Organization
Connector for WebEx (SAML 2.0)	□ A WebEx account. Trial accounts do not support federation.
Connector for Workday (SAML 2.0)	□ A Workday system administrator account.
Connector for Zoho (SAML 2.0)	□ A Zoho business account. □ A valid public domain that you have registered with Zoho. Select the **Enable MailHosting** option after logging in to the Zoho account.

13.3 Configuring Single Sign-On Connectors

After importing a single sign-on connector, you must configure the connector to work with the application. For more information about how to import connectors, see Chapter 4, “Importing Connectors from the Application Connector Catalog,” on page 43.

To configure a single sign-on connector:

1 Log in as an administrator to the CloudAccess administration console:

`https://`*`appliance_dns_name`*`/appliance/index.html`

2 Drag the appropriate connector from the **Applications** palette to the **Applications** panel.

3 On the **Configuration** tab, follow the on-screen prompts to configure the connector.

Use the information you obtained in the requirements section to configure the connector. For more information, see Section 13.2, “Requirements for the Single Sign-On Connectors,” on page 93.

4 Map the SAML Assertion attributes to the corresponding attributes in your identity source.

5 Expand the **Federation Instructions**, then copy and paste the instructions into a text editor. You will use this information to configure the federated connection for the CloudAccess appliance in the administration console.

NOTE: You must use a text editor that does not introduce hard returns or additional white space. For example, use Notepad instead of Wordpad.

6 Add an appmark for the connector to enhance the user experience. For more information, see Chapter 5, “Configuring Appmarks for Connectors,” on page 45.

7 Click **OK**, then click **Apply** to save the configuration.

8 Log in to your application account and use the **Federation Instructions** from Step 5 to configure the federated connection.

NOTE: Ensure that you include the beginning and end tags when you create the certificate from the **Federation Instructions**.

9 Perform policy mapping to specify entitlements for identity source groups. For more information, see “Mapping Authorizations” in the *CloudAccess Installation and Configuration Guide*.

13.4 Configuring ADFS to Connect to SharePoint

With additional configuration, the connector for ADFS allows users to single sign-on to SharePoint as well as ADFS.

13.4.1 Requirements

Verify that you meet the following requirements:

- ❐ One server with the following components installed:
  - ❐ Windows Server 2008 with the latest updates.
  - ❐ Active Directory with the latest updates.
  - ❐ ADFS 2.0 with the latest updates.
  - ❐ The SharePoint server connected to the ADFS server. Follow these instructions to connect the servers: How to Configure ADFS v 2.0 in SharePoint Server 2010 (http://technet.microsoft.com/en-us/library/hh305235%28v=office.14%29.aspx).
- ❐ Roles enabled within the SharePoint system using PowerShell scripts.
- ❐ A CloudAccess appliance installed and configured.

13.4.2 Modifying the Connector for ADFS Template

Use the Access Connector Toolkit to modify the definitions in the connector for ADFS.

1 Obtain a copy of the ZIP file for the connector for ADFS.

2 Log in as a CloudAccess administrator to the Access Connector Toolkit at

```
https://appliance_dns_name/css/toolkit
```

3 Click **Import**, browse to and select the connector’s ZIP file, then click **OK**.

4 Click the **Display Name** link for the connector to open it in the Edit Connector Template window.

5 Click the **Assertions** tab, then on the left side of the screen, click the **Attributes** tab.

6 Click **New,** then create a new Role attribute to use for the SharePoint connection.

6a Define the properties for the Role attribute:

Name: Specify `http://schemas.microsoft.com/ws/2008/06/identity/claims/role`.

Display Name: Specify `Role`.

Encoding: Leave this field blank.

Data Owner: Leave this field blank.

Default Value: Leave this field blank.

Required: Select **false** to make this attribute optional.

Description: Specify `A role assigned to the user account`.

Role Attribute: Select **true**, then continue to configure the role definitions.

6b Under **Roles**, click **New,** specify the following information, then click **Save**.

Name: Specify `ADMIN`.

Description: Specify `Administrator Role`.

6c Under **Roles**, click **New,** specify the following information, then click **Save**.

Name: Specify `USER`.

Description: Specify `User Role`.

6d Add or customize any additional roles that you need for the SharePoint environment, and save each one.

6e Click **Save** to save the Role attribute definition.

7 Click **Save** to apply the connector template changes.

8 Click the **Export** icon next to the **Display Name** for the connector template.

9 Save the ZIP file for use on this or another CloudAccess system.

10 Proceed to Section 13.4.3, "Importing the Modified Connector," on page 96.

13.4.3 Importing the Modified Connector

After you modify the connector for ADFS, you must import the connector into CloudAccess.

1 Log in as an administrator to the CloudAccess administration console at

```
https://appliance_dns_name/appliance/index.html
```

2 On the Admin page, click the **Tools** icon on the toolbar, then click **Import connector template**.

3 Click **Browse,** then browse to and select the ZIP file for the modified connector for ADFS.

4 Click **Import**.

The **Applications** palette displays the modified connector for ADFS.

5 Proceed to Section 13.4.4, "Configuring the Modified Connector," on page 97.

13.4.4 Configuring the Modified Connector

After you export and import the modified connector, you configure the connector by following the steps in Section 13.3, “Configuring Single Sign-On Connectors,” on page 94.

After you configure a connector for ADFS that supports SharePoint roles, you must modify ADFS and SharePoint to accept these roles. Proceed to Section 13.4.5, “Modifying Claim Rules in the ADFS System,” on page 97.

13.4.5 Modifying Claim Rules in the ADFS System

You must add ADFS claim rules between the CloudAccess appliance and ADFS. The purpose of these rules is to allow the user’s email address and the role to pass through to SharePoint.

To modify the claim rules:

1. Log in to your ADFS system.
2. Access the **Claims Provider Trusts** for the appliance.
3. Click **Edit Claim Rules**.
4. Add two rules in the Add Rule window using the following information:
   - Rule 1
     - **Claim rule template:** Select **Pass Through or Filter an Incoming Claim**.
     - **Claim rule name:** Specify `pass nameID`.
     - **Incoming claim type:** Specify `Name ID`.
     - **Incoming name ID format:** Specify `Email`.
     - **Pass through all claim values:** Select this option.
   - Rule 2
     - **Claim rule template:** Select **Pass Through or Filter an Incoming Claim**.
     - **Claim rule name:** Specify `pass Roles`.
     - **Incoming claim type:** Specify `Roles`.
     - **Pass through all claim values:** Select this option.
5. Exit the Rule editor.
6. Proceed to Section 13.4.6, “Configuring ADFS to Send SharePoint the Claim Rules,” on page 97.

13.4.6 Configuring ADFS to Send SharePoint the Claim Rules

The following steps map Email Address to Login on the SharePoint system, and send the user’s role. You have to perform these steps only once.

To send SharePoint the claim rules:

1. In the ADFS 2.0 console, click **Trust Relationships** > **Relying Party Trusts**.
2. Right-click *Name of your SharePoint system*, then select **Edit Claim Rules**.
3. Create two rules with the following information:
   - Rule 1
     - **Claim rule template:** Select **Transform an Incoming Claim**.

    - **Claim rule name:** Specify `NameID to EmailAddress`.
    - **Incoming claim type:** Specify `Name ID`.
    - **Incoming name ID format:** Specify `Email`.
    - **Outgoing claim type:** Specify `E-mail Address`.
    - **Pass through all claim values:** Select this option.
  - Rule 2
    - **Claim rule template:** Select **Pass Through or Filter an Incoming Claim**.
    - **Claim rule name:** Specify `pass Roles`.
    - **Incoming claim type:** Specify `Roles`.
    - **Pass through all claim values:** Select this option.
4. Exit the Rule editor.
5. Proceed to Section 13.4.7, "Configuring People Picker to Specify the Roles," on page 98.

13.4.7 Configuring People Picker to Specify the Roles

The default SharePoint People Picker configuration requires a repository of users and groups for the people picker to search. However, in a claims-based access model, the only information SharePoint has is the claims data associated with the current user's SAML assertion.

After you complete the ADFS configuration, you must configure the SharePoint 2010 option of **People Picker** to use the roles ADMIN and USER for claims received from ADFS. Before you begin, ensure that you have roles enabled within the SharePoint system.

To configure the People Picker:

1. Where the SharePoint 2010 system grants access, select **People Picker**.
2. Under ADFS, select **Role**.
3. In the **Find** field, specify either `ADMIN` or `USER`.

   This field must contain the name of the role you configure the connector to use in Section 13.4.2, "Modifying the Connector for ADFS Template," on page 95.
4. Select the role that SharePoint returns, then assign the role to the group within SharePoint.

14 Troubleshooting CloudAccess Connectors

Use the information in the following sections to troubleshoot any connector-related issues you might encounter.

- Section 14.1, “Using Troubleshooting Tools for Application Access Issues,” on page 99
- Section 14.2, “Troubleshooting Connector States,” on page 100
- Section 14.3, “Troubleshooting Provisioning Issues,” on page 101
- Section 14.4, “Troubleshooting Google Apps Issues,” on page 102
- Section 14.5, “Troubleshooting Salesforce Issues,” on page 103
- Section 14.6, “Troubleshooting Office 365 Issues,” on page 105
- Section 14.7, “Troubleshooting ServiceNow Issues,” on page 107
- Section 14.8, “Troubleshooting Failed Logins with Basic SSO Connectors,” on page 107
- Section 14.9, “Troubleshooting Other Connector Issues,” on page 107
- Section 14.10, “Troubleshooting Custom Connectors,” on page 108

14.1 Using Troubleshooting Tools for Application Access Issues

CloudAccess provides troubleshooting tools to help you resolve problems.

To access these tools:

1. Log in as an administrator to the CloudAccess administration console:

   ```
   https://appliance_dns_name/appliance/index.html
   ```

2. Under **Appliances**, click the node icon, then click **Enter troubleshooting mode**.
3. Click the node icon again, then click **Troubleshooting tools**.
4. Select one or more of the troubleshooting scenarios listed.
5. Duplicate the error or condition.
6. Click **Download CloudAccess Log Files** to download the logs.

After you obtain the logs, turn off troubleshooting mode by clicking the node icon again and then clicking **Exit troubleshooting mode**. Leaving the logs running affects the performance of your appliance.

All of the log files in Table 14-1 are included in the download, no matter what scenario you select. The scenario you select determines the amount of data displayed in the log files. Search the appropriate log file for errors while troubleshooting issues.

Table 14-1 Troubleshooting Log Files for Application Access Issues

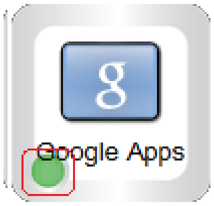
Feature	Logs
Identity Source Provisioning	bis_AD_<xxxxx>.log
	bis_AD_<xxxxx>_RL.log
	ConnectorLogs.txt
	bis_EDIR_h2q3p.log
	bis_EDIR_h2q3p_RL.log
Provisioning to the SaaS Applications	connectors_SFORCE_<xxxxx>_RL.log
	connectors_GOOGLEAPPS_<xxxxx>.log
	connectors_GOOGLEAPPS_<xxxxx>_RL.log
	connectors_O365_<xxxxx>.log
	connectors_O365_<xxxxx>_RL.log
	ConnectorLogs.txt
Mapping	RolesandResourceServiceDriver.log
	UserApplicationDriver.log
Approvals	jboss.log
Reporting	ManagedSystemGatewayDriver.log
	DataCollectionServiceDriver.log
Mobile Devices	mail
	mail.err
	mail.info
Custom Connectors	catalina.out
End User Authentication	catalina.out

14.2 Troubleshooting Connector States

CloudAccess displays indicators for the current state of the different appliance components. The display refreshes every five minutes. CloudAccess might not immediately display the change.

The health indicator is the small icon on each application connector in the **Applications** panel.

Figure 14-1 Application Health Indicator



The states are as follows:

Green: The connector to the application is healthy.

Yellow: The connector to the application contains warnings.

Red: The connector to the application contains errors or cannot communicate with the application.

Question mark: The connector to the application is in an unknown state.

Perform the following troubleshooting steps in the order listed:

1. Click **Show health** on the master node, then expand **Operational**, and check the status of **Provisioning**.

   If **Provisioning** is yellow or red, CloudAccess displays helpful information to help troubleshoot the issue.
2. Use the troubleshooting tools to gather logs, then look at the provisioning logs.
3. Make a cosmetic change to the application connector configuration, then click **Apply**.

   By forcing an **Apply**, the appliance refreshes the application connector state and this can resolve the issue.

14.3 Troubleshooting Provisioning Issues

Actions that are taken on users and groups in the identity source might not be reflected in the SaaS applications (Google Apps, Salesforce, ServiceNow, and Office 365). The following table lists the actions in the identity sources and the corresponding actions in the SaaS applications.

Table 14-2 *Provisioning Actions*

Identity Sources	SaaS Applications
Delete a user. (Or disable the user account.)	Disables the SaaS account. **NOTE:** In the MobileAccess app on an iOS device, the user continues to have access to the SaaS account until the in-progress user session times out.
Remove a user from the authorized group.	Disables the SaaS account.
Create a user.	◆ Creates an account for the user in the SaaS application, if the user is a member of a group with mapped SaaS authorizations. or ◆ Users are prompted to validate their information when they log in for the first time.
Move a user from out of the search context into the search context.	Creates an account for the user in the SaaS application, if the user is a member of a group with mapped SaaS authorizations.
Move a user out of the search context.	Disables the SaaS account.

By default, CloudAccess establishes identity based on an internal unique ID in the identity source, not based on the user name, and does not support recreating users with the same name unless they also have the same internal unique ID. After a user has been mapped and provisioned, if you delete the user from the identity source and then recreate that user with the same name, you will not be able to cache and activate the user in CloudAccess or provision the user to SaaS applications. When CloudAccess is unable to cache users properly, the Cached User Status Bar indicates this status with a lower number of active users than cached users.

IMPORTANT: CloudAccess does provide a **Relaxed user matching** option under **Advanced Options** on the configuration window for the identity source. If you select this option, CloudAccess matches users based on CN or sAMAccountName instead of the internal unique ID. This option enables you to recreate previously deleted users so CloudAccess can manage them again, but you must ensure that you do not create different users with the same CN or sAMAccountName as previously deleted users. Otherwise, those users will have access to the previously deleted users' cloud application data.

14.4 Troubleshooting Google Apps Issues

Use the information in the following sections to help you troubleshoot issues with the connector for Google Apps:

- Section 14.4.1, "Google Apps Users Can No Longer Log In After Enabling Single Sign-On," on page 102
- Section 14.4.2, "Users Are Not Provisioned to the Correct Organization," on page 103
- Section 14.4.3, "Chrome Profiles Cause Logins to Fail," on page 103
- Section 14.4.4, "Users Who Are Provisioned to Multiple Google Domains Cannot Access Original Mailbox," on page 103

14.4.1 Google Apps Users Can No Longer Log In After Enabling Single Sign-On

Issue: After you implement CloudAccess, you might have some issues with existing Google Apps for Business accounts. Any users who either do not exist in the identity source, or are not merged with the existing Google account, can no longer log in to the Google domain. For example, if user `jsmith` has an account in Google Apps for Business, and you implement CloudAccess with single sign-on, user `jsmith` cannot log in directly to the Google domain. Google Apps for Business does not allow both direct login and single sign-on to the domain.

Solution: Give users authorization to access the Google Apps for Business resource through CloudAccess.

1. (Conditional) If the matching account exists in Active Directory, skip to Step 2. Otherwise, create a matching account in the identity source (Active Directory).
2. Grant the user authorization to the Google Apps for Business resource by adding the user to the proper group in Active Directory. Alternatively, you can map the Active Directory group to the Google Apps for Business group through the Policy Mapping page. For more information, see "Loading Authorizations" in the *CloudAccess Installation and Configuration Guide*.

   The two accounts merge when the user receives authorization for Google Apps for Business through the Policy Mapping page. CloudAccess automatically generates a new password and resets the Google Apps for Business password. When users access the resource after the merge occurs, they automatically log in to Google Apps for Business through single sign-on.

14.4.2 Users Are Not Provisioned to the Correct Organization

By default, the connector for Google Apps places newly provisioned users into the top-level organization of your Google Apps domain. If you specified a sub-organization when you configured the connector for Google Apps, but users are still being provisioned to the top-level organization, verify that you entered a valid sub-organization in the **Default OrgUnit** field in the connector configuration. This field is free-form, is not case-sensitive, and is not validated. If you specify an invalid sub-organization, CloudAccess provisions the user to the top-level organization by default. If you have enabled tracing in the CloudAccess debugging tools, the `connector_GOOGLEAPPS_XXXXX.log` file will print a trace statement stating, "`Default OrgUnit configured on the connector does not exist in the Google Apps domain structure`" with the invalid value.

14.4.3 Chrome Profiles Cause Logins to Fail

Issue: If a user sets up a Chrome profile and then tries to use a Google Apps resource configured to use Chrome on a mobile device, 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 the MobileAccess app on the device. This issue occurs for any Google Apps resource (for example, Gmail or Google Drive) on iOS and Android mobile devices. (Bug 948622)

Workaround: 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.

14.4.4 Users Who Are Provisioned to Multiple Google Domains Cannot Access Original Mailbox

Issue: If you provision a user to multiple Google Apps domains and select the **Enable email proxy** option in the administration console, the user cannot open the mailbox for any domain except the last domain to which the user was provisioned. This issue occurs because the embedded mail proxy in the appliance uses an attribute from the user object that is single-valued, so it is set with the name of the last Google domain to which the user was provisioned.

Workaround: No workaround is available at this time.

14.5 Troubleshooting Salesforce Issues

Use the information in the following sections to help you troubleshoot issues with the connector for Salesforce:

- Section 14.5.1, "Salesforce Login Issues," on page 104
- Section 14.5.2, "Behavior of Service Provider-Initiated Login To Salesforce When Kerberos Is Enabled," on page 104

14.5.1 Salesforce Login Issues

Configuration of the connector for Salesforce may fail, even with valid credentials. One possible reason is that the Salesforce password has expired. Log in to the Salesforce site and reset your password. You receive a new password and a new security token. Use these credentials when creating the connector for Salesforce.

Even if your credentials are correct, you may occasionally be unable to log in to Salesforce, and the connector for Salesforce in CloudAccess may show an intermittent red status. Salesforce has API metering that limits the number of calls during a 24-hour period. For more information, see the following Salesforce resources:

- http://www.salesforce.com/us/developer/docs/api/Content/implementation_considerations.htm#sforce_api_rate_metering
- http://boards.developerforce.com/t5/General-Development/REQUEST-LIMIT-EXCEEDED/td-p/24901

If CloudAccess is configured with multiple nodes and the L4 switch uses load-balancing for transactions, the L4 switch must be configured to send transactions for a user's session to the same real server. A user might be unable to access Salesforce if the single sign-on request for its appmark is sent to a different real server than the user's login request to CloudAccess. For example, the same server might not be used if the L4 switch is set to use sticky-bit persistence and the user is logging in from a cookieless browser or mobile app. It can also happen if stickiness is not enabled on the L4 switch, or if the L4 switch does not support stickiness. If single sign-on is not working for the Salesforce appmark, you can use either of the following methods to ensure that requests for a user's session are sent to the same real server:

- Set the L4 switch to use IP-based persistence, which uses the user device's IP address to maintain an affinity between the user session and the same real server in the cluster. IP-based persistence can fail if a device's IP address changes between requests, such as if a user's mobile device changes networks when the user moves from one area to another.
- Use an identity-provider proxy approach that does not depend on the L4 switch configuration. This method can become chatty.

14.5.2 Behavior of Service Provider-Initiated Login To Salesforce When Kerberos Is Enabled

Issue: If you have Kerberos enabled on your CloudAccess cluster, service provider-initiated login attempts to Salesforce might result in the browser staying at the landing page after authenticating to CloudAccess instead of redirecting to Salesforce. This issue occurs only if Kerberos is enabled on the CloudAccess cluster. It occurs regardless of whether users log in with Kerberos single sign-on or with another authentication (for example, when the workstation is not a member of the Active Directory domain). (Bug 817909)

Workaround: This issue occurs on workstations running Windows 7 and Internet Explorer 9, but does not occur with Firefox on Windows 7.

You can prevent or address this issue by changing an option on the Single Sign-On Settings page at Salesforce. This page includes a radio button named **Service Provider Initiated Request Binding** with two options: **HTTP POST** (selected by default) and **HTTP Redirect**. If you have Kerberos enabled on your CloudAccess cluster, select **HTTP Redirect** instead of the default **HTTP POST** option. If you do not have Kerberos enabled on the CloudAccess cluster, you do not need to change this option.

14.6 Troubleshooting Office 365 Issues

Use the information in the following sections to help you troubleshoot issues with the connector for Office 365:

- Section 14.6.1, “Obtaining Installation and Provisioning Logs,” on page 105
- Section 14.6.2, “Office 365 Logout Error on Mobile Devices,” on page 105
- Section 14.6.3, “Display Name Does Not Change in Office 365 after Changing It in Identity Source,” on page 105
- Section 14.6.4, “Office Web Apps Cannot Be Assigned or Unassigned Without SharePoint Online,” on page 106
- Section 14.6.5, “Connectors for Office 365 that are Configured for Domain and Subdomains Do Not Work Correctly,” on page 106
- Section 14.6.6, “Installation Fails with Error: System.Runtime.InteropServices.COMException,” on page 106
- Section 14.6.7, “Renaming an Authorization for an Office 365 Account Requires Policy Remapping in CloudAccess,” on page 106

14.6.1 Obtaining Installation and Provisioning Logs

By default, CloudAccess does not generate an installation log when you install the connector for Office 365. If you want a log of the installation, you must launch the installer from the command line using the appropriate command. For more information, see Section 10.3, “Installing the Connector for Office 365,” on page 73.

The connector for Office 365 integrates with the Windows Event Log. The Windows Event Log displays the connector for Office 365 events as O365ConnectorEventLog. For more information about the Windows Event Log, see Windows Event Log (http://msdn.microsoft.com/en-us/library/windows/desktop/aa385780%28v=vs.85%29.aspx).

14.6.2 Office 365 Logout Error on Mobile Devices

If you configure Office 365 applications to launch with Safari on mobile devices, users are likely to encounter an issue when they try to log out of Office 365. When they tap the **Sign out** link at Office 365, they get the following Microsoft error: “`Sorry, but we're having trouble signing you out.`” If they go back to the MobileAccess app and tap the Office 365 appmark again, they get another Microsoft error: “`Sorry, but we're having trouble signing you in.`”

After this issue has occurred, the workaround for users to be able to use the Office 365 appmark again is to manually clear the cache and cookies in the Safari browser. However, if you want users to launch Office 365 in Safari, you can avoid this issue by having them set Safari’s cookie handling to “Never” block cookies. On iOS mobile devices this option is in the following location: **Settings** > **Safari** > **Privacy and Security** > **Block Cookies**.

14.6.3 Display Name Does Not Change in Office 365 after Changing It in Identity Source

Issue: If you change the display name of a user in Active Directory or eDirectory, the display name in Office 365 does not change accordingly. CloudAccess constructs the display name from the first and last name and does not synchronize the display name and full name from the identity source.

Workaround: Instead of changing the display name in the identity source, change the user's first and last name instead.

14.6.4 Office Web Apps Cannot Be Assigned or Unassigned Without SharePoint Online

When you assign or unassign Office 365 subscriptions to users, if you select Office Web Apps, you must also select SharePoint Online. This is a Microsoft Office 365 dependency, and the Office 365 admin portal page displays an error if you attempt to assign or unassign subscriptions without also selecting SharePoint Online. The Policy Mapping page in CloudAccess does not actually prevent you from assigning Office Web Apps by itself, but nothing happens and the logs show `Unable to assign this license`. In addition, if you assign several subscriptions to a user, and you include Office Web Apps but do not include SharePoint Online, none of the other licenses in that operation are applied until you add SharePoint Online. This behavior occurs on the Office 365 admin portal page as well as in CloudAccess.

14.6.5 Connectors for Office 365 that are Configured for Domain and Subdomains Do Not Work Correctly

Issue: If you configure a connector for Office 365 for a parent domain and then configure connectors for one or more child domains, users in the child domains do not see their assigned appmarks. Office 365 sends the same metadata for each domain, so the landing page shows only one of them. Users with policy mappings to the first connector installed can still see their appmarks.

Workaround: Microsoft does not support subdomains having different federated settings than their parent. To use a subdomain for Office 365, ensure that either you do not use Office 365 with the parent domain, or that both the parent domain and its subdomain have the identical federation settings.

14.6.6 Installation Fails with Error: System.Runtime.InteropServices.COMException

Issue: If you are installing the connector for Office 365 on a domain-joined server, but you are logged in as a local machine administrator, the installation fails with the error: `System.Runtime.InteropServices.COMException`

Solution: You must be logged in to the domain-joined server as a domain administrator or install the connector on a stand-alone Windows Server that is not part of the domain.

14.6.7 Renaming an Authorization for an Office 365 Account Requires Policy Remapping in CloudAccess

Issue: If an authorization at the Office 365 account is renamed, any existing policy mappings in CloudAccess are lost, because CloudAccess uses the account name for policy mapping rather than the underlying static ID of the authorization. If you rename an authorization in Office 365, CloudAccess sees the action as a delete and create, and removes any existing policy mappings for the authorization.

Workaround: After changing the authorization name in Office 365, use the Policy page to re-map entitlements for the renamed authorization, and then use the Approval page to re-approve, if necessary.

14.7 Troubleshooting ServiceNow Issues

Issue: The user account exists in ServiceNow. The end user can log into CloudAccess, and sees the ServiceNow "Home" appmark. When the end user clicks the appmark, a new tab opens for the ServiceNow login, but the user sees a brief message of `User not found`, and is then redirected to the ServiceNow logout page.

Workaround: This issue can occur if the ServiceNow identity provider **User** field and the connector **Assertion Naming** option do not match. For example, one is set to `user_name` and the other is set to `Email Address`. They must match for the SAML assertion to find the proper account.

14.8 Troubleshooting Failed Logins with Basic SSO Connectors

If your users are unable to log in to sites using the Basic SSO apps or plug-ins, have the users delete the credentials stored for that site. Users can delete their own credentials from the landing page.

This problem arises when a site uses two different HTML forms or changes fields on the same HTML form during the login or password change process.

Having the users delete their stored credentials from the appliance allows the Basic SSO app or the plug-in to save the new credentials.

14.9 Troubleshooting Other Connector Issues

Use the information in the following sections to help you troubleshoot other connector issues:

- Section 14.9.1, “Logging Out of Identity Provider Landing Page Does Not Result in Logging Out of SaaS Accounts,” on page 107
- Section 14.9.2, “Admin Page Does Not Provide a Way to View SaaS Metadata,” on page 107
- Section 14.9.3, “User Session Timeout Affects Box Connector,” on page 108

14.9.1 Logging Out of Identity Provider Landing Page Does Not Result in Logging Out of SaaS Accounts

Issue: Logging out of the landing page might not result in logging out of the SaaS accounts, depending on support and configuration for SAML Single Logout at the SaaS provider. Many SaaS providers do not support the SAML Single Logout service. The same issue exists with service provider-initiated logouts.

Workaround: Close the browser to allow the abandoned browser session to time out, so the session cannot be accessed again.

14.9.2 Admin Page Does Not Provide a Way to View SaaS Metadata

Issue: The Admin page in CloudAccess does not currently provide a means of viewing the critical content in an uploaded metadata file, such as when you configure the connector for Salesforce.

Workaround: No workaround is available at this time. Since metadata for connectors must be unique, ensure that the metadata file is correct before uploading it.

14.9.3 User Session Timeout Affects Box Connector

Issue: If you set the user session timeout for the cluster to 75 minutes or longer, the Box connector displays an error when users attempt to use single sign-on to Box. (Bug 814752)

Workaround: To ensure that single sign-on works for the Box connector, set the User session timeout value to 74 minutes or less. This is a cluster-level setting so it will affect behavior of user sessions not using Box as well.

14.10 Troubleshooting Custom Connectors

Custom connectors allow for authentication into other systems, but they do not provide provisioning of user accounts. Unlike the connector for Salesforce, CloudAccess does not create a specific log for each custom connector.

CloudAccess captures all information about custom connectors in the `catalina.out` file. To troubleshoot issues with custom connectors and capture the information in the `catalina.out` file, perform the following steps:

1. Log in as an administrator to the CloudAccess administration console:

   https://*appliance_dns_name*/appliance/index.html

2. Under **Appliances**, click the node icon, then click **Enter troubleshooting mode**.
3. Click the node icon again, then click **Troubleshooting tools**.
4. Select **Authentication / Single Sign-on** to increase the logging levels.
5. Duplicate the error or condition.
6. Click **Download CloudAccess Log Files** to download the logs.
7. Extract the download file and search for `catalina.out`.
8. Open `catalina.out` in a text editor, then search for errors in association with your custom connector.

A Custom Connector Worksheets

CloudAccess provides the Access Connector Toolkit (ACT) that allows you to create custom connectors. If you need help creating a custom connector to use with CloudAccess, with Priority Support you have the option to open a service request with Technical Support (http://www.netiq.com/support). Technical Support is available to provide toolkit support as well as to configure the connectors to work with integrated applications. Additional information from the SaaS provider is usually required.

Before you contact Technical Support, please complete the appropriate worksheet for the connector type that you want to create. The more information you can provide, the better and quicker Technical Support can help you create the connector.

- Section A.1, "Worksheet for SAML or WS-Federation Custom Connectors," on page 109
- Section A.2, "Worksheet for SAML In Custom Connectors," on page 110
- Section A.3, "Worksheet for Basic SSO Custom Connectors," on page 111

A.1 Worksheet for SAML or WS-Federation Custom Connectors

For a SAML or WS-Federation custom connector, the destination service provider for the application is the trusted partner. Each connector requires information about how they support federation for the SAML protocol or WS-Federation protocol.

Table A-1 *Worksheet for a SAML or WS-Federation Custom Connector*

	Gather the following information:
□	Which federation specifications will be used with various trusted partners? □ WS-Federation □ SAML 2.0
□	Is the metadata (SAML or WS-Federation) from the trusted partner available?
□	What profiles will you use to federate with your partners? □ WS-Federation Passive Requestor profile □ Browser POST profile □ Browser Artifact profile
□	Is encryption of the assertions required? If so, which transport security protocols and certificates will be used?
□	What user information is required by your partner for SSO? For example: email address, CN, and so on.

	Gather the following information:
□	What name identifier format does your partner expect? □ Persistent □ Transient □ Email address □ Unspecified
□	What attributes are required by your partner? Does a sample assertion exist from the trusted partner?
□	To what URL on the partner side should an assertion or a claim be sent? (Assertion Consumer Service URL)
□	To what URL on the partner side should a logout request be sent? (Logout URL and/or Logout Response URL)
□	Do users need to be redirected to a specific application URL after an assertion has been successfully validated? (Destination URL)
□	What are the contact details for the trusted partner (or partners), should we need to get them involved?
□	All information needed by the trusted partner is available via the metadata at `https://`*`appliance_dns_name`*`/osp/a/t1/auth/saml2/metadata`

A.2 Worksheet for SAML In Custom Connectors

For a SAML Inbound (SAML In) custom connector, the identity provider is the trusted partner. Each connector requires information about how they support SAML federation.

Table A-2 *Worksheet for a SAML Inbound Custom Connector*

	Gather the following information:
□	Which federation specifications will be used with various trusted partners? □ SAML 2.0 □ SAML 1.*x*
□	Is the SAML metadata from the trusted partner available?
□	What profiles will you use to federate with your partners? □ Browser POST profile □ Browser Artifact profile
□	Which transport security protocols and certificates will be used? Assertions must be signed, and may be encrypted.
□	What user information does the partner send for SSO? For example: email address, CN, and so on.

	Gather the following information:
□	What name identifier format does your partner send with an assertion? □ Persistent □ Transient □ Email address □ Unspecified
□	What attributes does your partner send? Does a sample assertion exist from the trusted partner?
□	To what URL on partner side should a logout request be sent? (Logout URL and/or Logout Response URL)
□	What are the contact details for the trusted partner (or partners), should we need to get them involved?
□	All information needed by the trusted partner is available via the metadata at `https://appliance_dns_name/osp/a/t1/auth/saml2/metadata`

A.3 Worksheet for Basic SSO Custom Connectors

Each Basic SSO connector requires information about the HTML forms-based login for an application. A Fiddler trace output of a successful login to the application will include all of the information you need to complete the worksheet for a Basic Single Sign-On custom connector. For more information about Fiddler, see the *Fiddler Web Debugging Tool* on the Microsoft Developer Network website.

Table A-3 *Worksheet for a Basic SSO Custom Connector*

	Gather the following information:
□	What are the HTML login page details? □ Domain URL of the web service or application □ Domain URL of the login page for the web service or application □ Form ID or name for the user name □ Form ID or name for the user password □ Input type for the form (button, image, string)
□	On what domain is the form?