Contents

Prerequisites 2

Configuring CloudAccess to Provide Identity Services to Access Manager 4

Configuring Access Manager to Use CloudAccess as an Identity Provider 5

Configuring Identity Provider-Initiated Logins 17

Troubleshooting the CloudAccess and Access Manager Configuration 17

Using CloudAccess as a Trusted Identity Provider for Access Manager

Technical Reference

May 8, 2014

This document provides information about integrating NetIQ CloudAccess 2.0 and NetIQ Access Manager 4.0 to provide cloud and mobile authentication to applications in your network. It also provides troubleshooting tips to help resolve common issues.

You can use CloudAccess to provide trusted identity services to NetIQ Access Manager. Although you install the Access Manager Identity Server as an identity provider (IdP), you can also configure it to be a service provider (SP) that consumes authentication information. CloudAccess provides federation and single sign-on services to Access Manager, but not provisioning. You can also configure CloudAccess to provide MobileAccess features for Access Manager users.

To establish a secure exchange of authentication information, you must configure a trust relationship between CloudAccess and Access Manager. Until this two-way trust is established, federation cannot occur. CloudAccess provides an Access Manager connector that uses the Security Assertion Markup Language (SAML) 2.0 protocol for federation and single sign-on services. This protocol ensures the secure exchange of authentication and attribute information about users. You must also configure Access Manager to consume the information sent by the connector.

Legal Notice

Prerequisites

- ☐ A CloudAccess appliance, installed and configured. MobileAccess configuration is optional, depending on your user authentication needs.
- ☐ A NetIQ Access Manager system, installed and configured.

  Ensure that SSL communications are enabled for Identity Server and Access Gateway, and that both components are configured to trust the same signing certificate authority. For more information, see "Enabling SSL Communications" (https://www.netiq.com/documentation/netiqaccessmanager4/basicconfig/data/b6vcbhk.html) in the *NetIQ Access Manager 4.0 Setup Guide* (https://www.netiq.com/documentation/netiqaccessmanager4/basicconfig/data/bookinfo.html). You will use this signing certificate for the Access Manager connector in CloudAccess.
- ☐ Access Manager user accounts for each user who wants the single sign-on service.
- ☐ The metadata file from your Access Manager system for SAML 2.0 services:

  ```
  https://<nam_identity_server_DNS_name>/nidp/saml2/metadata
  ```

☐ The SSL signing certificate from Access Manager.

IMPORTANT: The configuration in this section assumes that you have configured SSL communications for Access Manager. The SSL signing certificate does not necessarily need to come from an external certificate authority, but you must use the same certificate for the Access Manager connector in CloudAccess when you set up the federation. Each provider must trust the SSL certificate authority.

For information about configuring SSL communications for Access Manager, see "Security and Certificate Management" (https://www.netiq.com/documentation/netiqaccessmanager4/adminconsolehelp/data/b3trhnr.html) in the *NetIQ Access Manager 4.0 Administration Console Guide* (https://www.netiq.com/documentation/netiqaccessmanager4/adminconsolehelp/data/bookinfo.html).

SSL is used for the secure exchange of authentication information between CloudAccess and Access Manager. When you configure the Access Manager connector in CloudAccess, you must import the trusted root certificate from the Access Manager NIDP Trust Store. Failure to import the certificate causes numerous system errors.

You can download the certificate from the Trusted Roots configuration for Access Manager. Store the file in a location that you can browse to from the CloudAccess appliance.

1. In the Access Manager Administration Console, click **Devices > Identity Servers > ClusterName > Security > Trusted Roots**.
2. Click the signing certificate name.
3. On the Certificate Details page, select **Export Public Certificate**, then click **PEM** as the file type.

   A PEM-encoded file is a Base64-encoded DER certificate that is enclosed between `BEGIN CERTIFICATE` and `END CERTIFICATE` tags.
4. Store the **PEM** file in a location that you can browse to from the CloudAccess appliance when you configure the connector for Access Manager.

You can alternatively copy the certificate information from the `ds:X509Certificate` field in the Access Manager metadata file. Ensure that you add `-----BEGIN CERTIFICATE-----` before the encoded information, and add `-----END CERTIFICATE-----` after the encoded information.

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

☐ If you use an eDirectory identity source for Access Manager and you need to provide access to Access Gateway protected resources that require a user name and password, you must enable Universal Password in eDirectory for the Access Manager LDAP connection.

NOTE: Universal Password Retrieval options must be properly set in the configuration of the Universal Password policy in eDirectory, so that it allows the password to be retrieved from the Access Manager user store.

For more information, see *Unable to retrieve Universal Password from eDirectory using PasswordFetchClass (TID 7007114)* (http://www.novell.com/support/kb/doc.php?id=7007114).

Configuring CloudAccess to Provide Identity Services to Access Manager

To provide identity services to Access Manager, CloudAccess and MobileAccess must trust Access Manager as a service provider. Establish this trust by enabling and configuring the NetIQ Access Manager connector.

To enable the Access Manager connector and configure it to connect to your Access Manager system:

1. Before you begin, download the metadata file for SAML 2.0 services from your Access Manager system:

   `https://<access_manager_identity_server_DNS_name>/nidp/saml2/metadata`

   You need information from this file to configure the Access Manager connector.
2. In the CloudAccess administration console, drag the Access Manager connector from the Applications Palette and drop it in the Applications pane.
3. Select the Access Manager connector, then click **Configure**.
4. On the Configuration window, use information from the Access Manager metadata file to specify the following connector settings.

   NOTE: The information from the Access Manager metadata file is case sensitive.

Connector Parameter	Value	Metadata Parameter or Description
Display name	*my_nam_sp*	Specify a unique name for your Access Manager service provider.
Assertion consumer service URL	`https://`*`idp.example.com`*`/nidp/saml2/spassertion_consumer`	Specify the location in the `AssertionConsumerService` section for HTTP-POST bindings.
Destination URL	*`https://web_redirect_url`*	(Optional) After a successful authentication by the IdP, the web browser is redirected to the secure destination URL.
EntityID	`https://`*`idp.example.com`*`/nidp/saml2/metadata`	`entityID` Ensure that you specify the ID with lowercase characters.
Logout response URL	`https://`*`idp.example.com`*`/nidp/saml2/spslo_return`	Specify the response location in the `SingleLogoutService` section for HTTP-POST bindings.
Logout URL	`https://`*`idp.example.com`*`/nidp/saml2/spslo`	Specify the location in the `SingleLogoutService` section for HTTP-POST bindings.
Signing certificate		Browse to and select the file that contains the Access Manager SSL certificate.

5. In the **Assertion Attribute Mappings** section, select an attribute from the **NameID** list to use for mapping users in the federation.

   Specify the LDAP attribute that contains a user's name identifier in the Access Manager user store. CloudAccess and Access Manager can use different user stores, as long as you can find an attribute that is consistent between them.

   For example, select **X-Custom1**, where you have created a custom mapping of the employee ID attribute to the X-Custom1 attribute in the CloudAccess identity source.

6. Expand **Federation Instructions**, then copy and paste the instructions into a text file to use during the Access Manager configuration.

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

7. Click **OK** to save the settings.
8. Click **Apply** to enable the connector.
9. Wait for the updates to take effect on all nodes in the cluster.
10. Continue with "Configuring Access Manager to Use CloudAccess as an Identity Provider" on page 5.

Configuring Access Manager to Use CloudAccess as an Identity Provider

Use the instructions in this section to configure Access Manager to use CloudAccess or MobileAccess as a trusted external identity provider.

IMPORTANT: For simplicity, the instructions in this section refer to the CloudAccess appliance as the external identity provider. The configuration also applies for MobileAccess users.

- Creating an Attribute Set to Use for the Identity Provider Attributes (page 5)
- Creating an Attribute Matching Expression to Use for the Identity Provider User Identification (page 6)
- Creating an External Identity Provider in Access Manager (page 7)
- Creating an External Authentication Contract for the Identity Provider (page 7)
- Configuring a SAML 2.0 Authentication Request for the Identity Provider (page 10)
- (Optional) Configuring Password Retrieval (page 12)
- Configuring a User Identification Method for the Identity Provider (page 15)
- Configuring Attributes for the Identity Provider (page 16)
- Assigning the External Authentication Contract to Protected Resources (page 16)

Creating an Attribute Set to Use for the Identity Provider Attributes

Access Manager uses attribute sets to provide a common naming scheme for the exchange of authentication information. Using an attribute set reduces the traffic between the identity provider and the service provider's LDAP server, because the attribute information can be gathered in one request at authentication rather than in a separate request for each attribute when a policy or protected resource needs the attribute information.

You map an LDAP attribute set in the Access Manager user store, such as `givenName`, to the equivalent attribute used in the authentication information. The authentication attribute name does not necessarily match the attribute name in the CloudAccess identity source or the LDAP attribute in the Access Manager user store. For example, the CloudAccess identity source might use the `workforceID` attribute, but call the attribute `NameID` in the authentication information. You want to map the latter (`NameID`) to the appropriate Access Manager LDAP attribute. The attribute set can then be used for policy enforcement, user identification, and data injection.

1. In the Access Manager Administration Console, click **Devices > Identity Server > Shared Settings > Attribute Sets > New**.
2. Specify a unique name for the attribute set, then click **Next**.
3. Click **New** to add an attribute mapping to the set.
4. In the **Add Attribute Mapping** window, specify the following information:

Parameter	Value	Description
Local attribute	*`Ldap Attribute:workforceID [LDAP Attribute Profile]`*	Select the LDAP attribute in the Access Manager user store that you want to match.
Remote attribute	`NameID`	This is the attribute presented by CloudAccess.
Remote namespace	**none**	This value allows CloudAccess as the identity provider to use a default namespace.
Remote format	**unspecified**	The interpretation of the content is implementation-specific.

5. Click **OK**.

   On the Mapping page, the system displays the map settings for the remote attribute NameID.
6. Click **OK** to map and save the attribute set.
7. Continue with "Creating an Attribute Matching Expression to Use for the Identity Provider User Identification" on page 6.

Creating an Attribute Matching Expression to Use for the Identity Provider User Identification

1. In the Access Manager Administration Console, click **Devices > Identity Servers > Shared Settings > User Matching Expressions**.
2. Click **New**, then specify a unique name for the attribute matching expression.
3. Under **User Matching Expression**, click **New Logic Group**.
4. Under the new group, click the **Add Attributes** icon (plus sign), select the LDAP attribute to add to the logic group, then click **OK**.

   Use the same local LDAP attribute that you specified in the attribute set, such as `Ldap Attribute:workforceID`.
5. Accept the default **Type** setting.

   For CloudAccess, there is only one group and one attribute, so the Type setting has no impact.

   The **Type** specifies the `AND` or `OR` boolean condition for matching that applies between groups used in a user-matching expression. Attributes within a group are always the opposite of the **Type** condition. For example, if the **Type** is `AND`, the attributes within the group are matched as `OR` conditions.

6 Click **OK** to save the attribute expression.

7 Continue with "Creating an External Identity Provider in Access Manager" on page 7.

Creating an External Identity Provider in Access Manager

In Access Manager, create a new identity provider for the CloudAccess appliance:

1 In the Access Manager Administration Console, click **Devices** > **Identity Servers** > **[ClusterName]** **> SAML 2.0**.

2 On the Trusted Providers page, click **New**, then click **Identity Provider**.

3 Specify a unique name to use for the identity provider that represents your CloudAccess appliance.

4 In the **Source** list, select one of the following methods to specify the Source to use for metadata about the CloudAccess appliance:

- **Metadata Text:** Open the metadata file in a web browser, copy the entry for Access Manager, then paste the information in the **Metadata Text** field. The URL is

  `https://`*`appliance`*`:443/osp/a/t1/auth/saml2/metadata`

- **Metadata URL:** Specify the metadata URL. The URL is

  `https://`*`appliance`*`:443/osp/a/t1/auth/saml2/metadata`

- **Metadata Repositories:** Select the repository name from the **Repository** field.

  You can use this option only if you created a metadata repository in **Shared Settings** for the CloudAccess metadata. For more information, see "Metadata Repositories" (https://www.netiq.com/documentation/netiqaccessmanager4/identityserverhelp/data/metadatarepositories.html) in the *NetIQ Access Manager Identity Server Guide* (https://www.netiq.com/documentation/netiqaccessmanager4/identityserverhelp/data/bookinfo.html).

5 Review the entities that were imported for SAML 2.0 from the CloudAccess metadata file.

Typically, the entities are Unspecified Federation and Post binding. No contracts and attributes are set at this time.

6 On the Metadata page, review the signing certificate by expanding **KeyInfo > X509Data > X509Certificate**.

7 Click **Finish** to save the new provider.

The system displays the trusted provider on the SAML 2.0 page.

8 Click **OK**.

9 Update the Access Manager Identity Server.

10 Continue with "Creating an External Authentication Contract for the Identity Provider" on page 7.

Creating an External Authentication Contract for the Identity Provider

Access Manager uses an authentication contract to define how authentication occurs. The contract defines a string that the identity provider uses to match an incoming authentication request from Access Manager. You can assign a contract to one or more resources.

An external authentication contract allows you to use CloudAccess as the primary authentication method for a resource. The contract can allow users to authenticate only through CloudAccess, or to alternatively authenticate through local contracts of equal or higher authentication levels.

IMPORTANT: If the protected resources are authenticated primarily by a local contract, but might alternatively be authenticated by CloudAccess, you can modify the local contract you want to use to allow it to be satisfiable by an external provider.

For more information about external authentication contracts in Access Manager, see "Configuring Authentication Contracts" (https://www.netiq.com/documentation/netiqaccessmanager4/identityserverhelp/data/localcontract.html) in the *NetIQ Access Manager Identity Server Guide* (https://www.netiq.com/documentation/netiqaccessmanager4/identityserverhelp/data/bookinfo.html).

To define an authentication contract for CloudAccess:

1 In the Access Manager Administration Console, click **Devices > Identity Servers > Edit > Local > Contracts**.

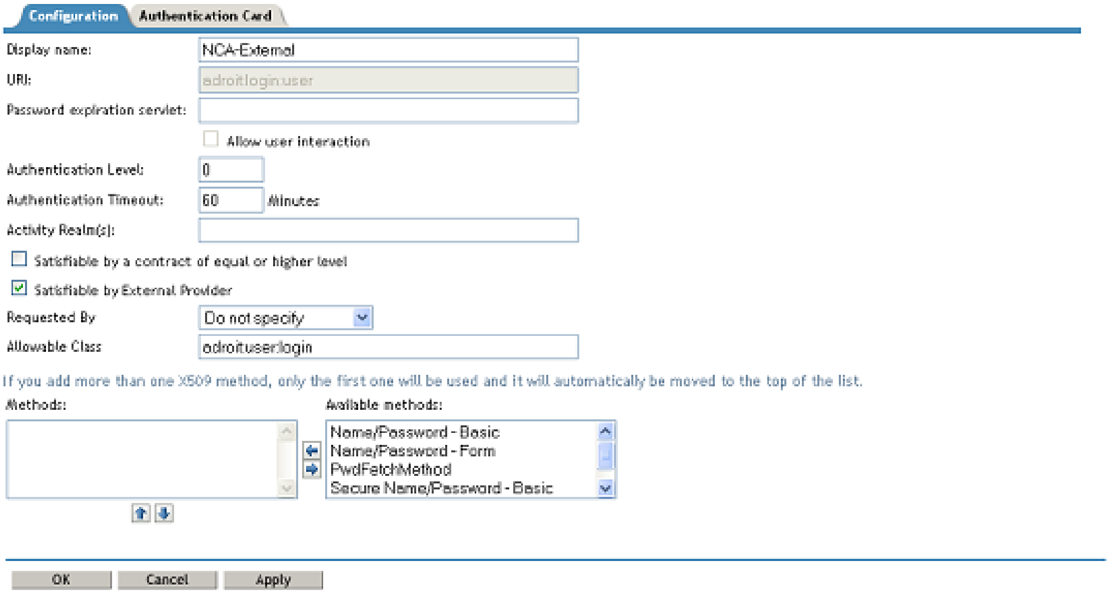
2 Click **New**.

3 On the Configuration page, define an external authentication contract that matches the authentication type that the CloudAccess appliance requires.

For example, specify the following fields to create a contract named `NCA-External`:

Parameter	Value	Description
Display name	*NCA-External*	Specify a unique name for this contract.
URI	`adroit:user:login`	The URI matches the requested class type.
Password expiration servlet	Leave the field blank.	Do not use this setting for an external identity provider.
Allow user interaction	Deselect the check box.	Do not use this setting for an external identity provider.
Authentication level	0	The identity provider is the trusted source for authentication.
Authentication timeout	60	Recommended. A session times out if it is inactive for the specified number of minutes. The value can be from 5 minutes to 66535 minutes and must be divisible by 5. You can specify a value to meet your security requirements. Shorter timeout periods generate more authentication traffic.
Activity realm	Leave the field blank.	Do not use this setting for an external identity provider.

Parameter	Value	Description
Satisfiable by a contract of equal or higher level	Deselect the check box.	Deselect the check box to make CloudAccess the only trusted identity provider for authentication. This setting does not allow access through local authentication. Select the check box to allow the authentication to also be satisfiable by local contracts of an equal or higher authentication level.
Satisfiable by external provider	Select the check box.	This option allows the contract to be selected when you configure an identity provider for the Liberty or SAML 2.0 protocol.
Requested by	**Do not specify**	This value allows the identity provider to send any type of authentication to satisfy a service provider's request. The service provider cannot send a request for a specific authentication type or contract.
Allowable class	`adroit:user:login`	This value matches the CloudAccess authentication type.
Methods and Available methods	Leave the **Methods** list blank.	Not applicable for an external identity provider.



Display name: NCA-External

URI: adroit:login:user

Password expiration servlet:

☐ Allow user interaction

Authentication Level: 0

Authentication Timeout: 60 Minutes

Activity Realm(s):

☐ Satisfiable by a contract of equal or higher level

☑ Satisfiable by External Provider

Requested By: Do not specify

Allowable Class: adroit:user:login

If you add more than one X509 method, only the first one will be used and it will automatically be moved to the top of the list.

Methods:

Available methods:

- Name/Password - Basic
- Name/Password - Form
- PwdFetchMethod
- Secure Name/Password - Basic

4 Click **Next**.

5 On the **Authentication Card** tab, configure an authentication card for the contract:

Parameter	Value	Description
ID	*NCA-External*	Specify an alphanumeric value that identifies the card.
Text	*NCA-External*	Specify the text that is displayed on the card to the user.
Image	**Customizable**	(Optional) You can specify an image set to use for this contract.
Show Card	Select the check box.	The card is shown to the user, which allows the user to select and use the card for authentication.
Passive Authentication Only	Select the check box.	The Access Manager Identity Server will not prompt the user for credentials. If the Identity Server can fulfill the authentication request without any user interaction, the authentication succeeds. Otherwise, it fails.
Authentication contracts	**Secure Name/Password - Form** **Name/Password - Form**	Use the arrows to move the required contracts from the **Available contracts** list to the **Satisfies contract** list. This creates a mapping between the external provider class reference and local authentication contracts.

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

7 Update the Access Manager Identity Server:

  7a Click **Devices**, then click your Identity Server.

  7b Click **Update All**, then click **OK**.

  7c Wait for Access Manager to process the new configuration.

8 Continue with "Configuring a SAML 2.0 Authentication Request for the Identity Provider" on page 10.

Configuring a SAML 2.0 Authentication Request for the Identity Provider

Access Manager uses an authentication request to define the federation method and the authentication contract to use for an external identity provider. This relationship between the identity provider and service provider enables single sign-on and single log-out. To enable the authentication process for CloudAccess, you must create an authentication request that uses the external authentication contract that you created for it. The authentication type in the contract must match the string that the service provider sends in an authentication request.

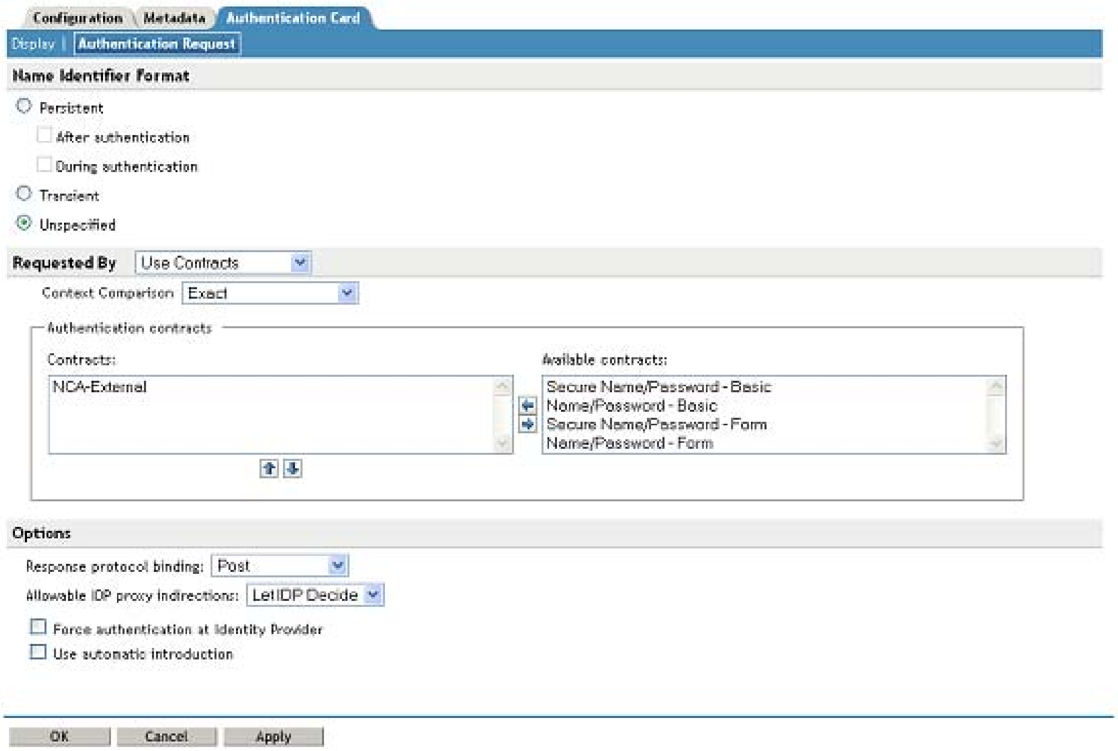
For more information about authentication requests in Access Manager, see "Configuring an Authentication Request for an Identity Provider" (https://www.netiq.com/documentation/netiqaccessmanager4/identityserverhelp/data/bm9y1rw.html) in the *NetIQ Access Manager Identity Server Guide* (https://www.netiq.com/documentation/netiqaccessmanager4/identityserverhelp/data/bookinfo.html).

To define a SAML 2.0 authentication request for CloudAccess:

1. In the Access Manager Administration Console, click **Devices > Identity Servers > Edit > SAML 2.0 > [Identity Provider] > Authentication Card** > **Authentication Request.**
2. On the Authentication Request page, modify the authentication request card for the CloudAccess appliance.

   For example, specify the following fields to use a contract named `NCA-External`:

Parameter	Value	Description
Name Identifier Format	**Unspecified**	Either a persistent identifier or a transient identifier can be sent in an authentication request. An identifier federates the user profile on the identity provider with the user profile on the service provider. A persistent identifier remains intact between sessions. A transient identifier expires between sessions.
Requested By	**Use Contracts**	An authentication contract must be used.
Context Comparison	**Exact**	The identity provider uses the URI in the request to find an authentication procedure. If it finds an exact URI match, the identity provider prompts the user for the appropriate credentials. If it does not find an exact match, the user is denied access.
Authentication Contracts	*NCA-External*	Specify the contract that you defined in "Creating an External Authentication Contract for the Identity Provider" on page 7.
Response protocol bindings	**Post**	Use the post method for transmitting assertions between the authenticating system and the target system.
Allowable IDP proxy indirections	**Let IDP Decide**	This value allows the trusted identity provider to decide how many times a request can be proxied.
Force authentication at Identity Provider	Deselect the check box.	The identity provider does not prompt users for authentication if they are already logged in.
Use automatic introduction	Deselect the check box.	Automatic introduction is not used for single sign-on.



Name Identifier Format

○ Persistent

☐ After authentication

☐ During authentication

○ Transient

● Unspecified

Requested By Use Contracts

Context Comparison Exact

Authentication contracts

Contracts: NCA-External

Available contracts: Secure Name/Password - Basic; Name/Password - Basic; Secure Name/Password - Form; Name/Password - Form

Options

Response protocol binding: Post

Allowable IDP proxy indirections: Let IDP Decide

☐ Force authentication at Identity Provider

☐ Use automatic introduction

3 Click **OK** twice.

4 Update the Access Manager Identity Server.

5 If you have Access Gateway protected resources that require a password and you do not have a password retrieval method defined for Access Manager, continue with "(Optional) Configuring Password Retrieval" on page 12. Otherwise, continue with "Configuring a User Identification Method for the Identity Provider" on page 15.

(Optional) Configuring Password Retrieval

The identity provider contract for CloudAccess does not use a user name and password for the credentials. To allow single sign-on to Access Gateway protected resources that require a user's name and password, you must configure the PasswordFetchClass to retrieve them. You create the class, then create a password retrieval authentication method from the class.

NOTE: MobileAccess cannot send a user's password for a proxy application to the back-end web service. However, the password-retrieval method specifies a static string that is accepted for all users.

The service provider executes the password retrieval after the identity provider completes the remote authentication and federation. It stores the user name and password with the LDAP credentials, then allows the additional user-specific attributes to be injected in SAML assertions for authentication sent to and consumed by the Access Gateway that protects the back-end resources. This advanced authentication enables users to access the back-end protected resources.

IMPORTANT: The PasswordFetchClass works only with eDirectory user stores where Universal Password is enabled.

1. Create a password fetch class in Access Manager:
   - **1a** In the Access Manager Administration Console, click **Devices** > **Identity Servers** > **Edit** > **Local** > **Classes**.
   - **1b** Click **New**.
   - **1c** On the General page, configure the following settings:

Parameter	Value	Description
Display name	*`Password Retrieval`*	Specify a unique name for the authentication class.
Java class	**PasswordFetchClass**	Select this value from the available Java classes. The Java class path is configured automatically.

   - **1d** Click **Next**.
   - **1e** On the **Properties** tab, configure the following **General** settings and **Userstore Lookup** settings:

Parameter	Value	Description
Ignore password retrieval failure	Deselect the check box.	Deny access for a user if the user's password cannot be retrieved.
Retain previous principal	Select the check box.	Retain the principal obtained from the previous authentication method.
Password to be retrieved	**Universal Password**	If your users have been configured to use a universal password, select **Universal Password**. Otherwise, select **Simple Password**.
User lookup method	**Based on the CN of the user object**	The CN of the user objects are mapped between two different user stores to allow for password retrieval.

   - **1f** Click **OK**.
2. Create a method for the password fetch class:
   - **2a** In the Access Manager Administration Console, click **Devices > Identity Servers > Edit > Local > Methods.**
   - **2b** Click **New**.

2c Configure the following settings for the new authentication method:

Parameter	Value	Description
Display name	`Password Retrieval`	Specify a meaningful name for the authentication class.
Class	Password Retrieval	Specify the password fetch class name that you created in Step 1c.
Identifies User	Select the check box.	If you enable this option on just one method in the contract, that method identifies the user when the authentication method succeeds.
Overwrite Temporary User	Select the check box.	Real user credentials retrieved with this method overwrite the temporary user credentials from a prior authentication method in the same session.
Overwrite Real User	Select the check box.	Real user credentials retrieved with this method overwrite the real user credentials from a prior authentication method in the same session.
User Stores	User stores you want to use with this authentication method	Use the arrows to select the user stores from the available user stores. If you have several user stores, the system searches through them based on the order specified here. If a user store is not moved to the **User stores** list, users in that user store cannot use this method for authentication.
Properties	(Optional)	You can specify properties only for classes that have properties to control. The properties allow you to create different methods that are customized for different uses.

2d Click **Finish**.

3 Click **Apply**.

4 Update the Access Manager Identity Server.

5 Continue with "Configuring a User Identification Method for the Identity Provider" on page 15.

Configuring a User Identification Method for the Identity Provider

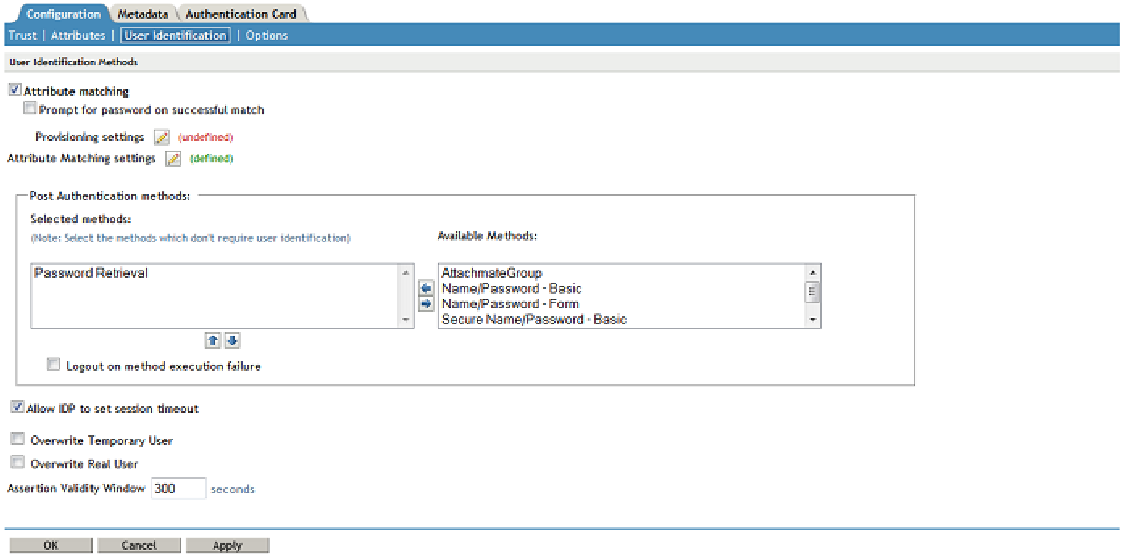
During the authentication, CloudAccess matches the user with an account in the Access Manager user store. The matching process allows CloudAccess to retrieve information about the user, such as the name, email, roles, and so on. You must specify the user identification method that is used to match the user account at the identity provider (CloudAccess) with a user account at the service provider (Access Manager).

For more information about user identification in Access Manager, see "Selecting a User Identification Method for Liberty or SAML 2.0" (https://www.netiq.com/documentation/netiqaccessmanager4/identityserverhelp/data/bmmudo8.html#userident) in the *NetIQ Access Manager Identity Server Guide* (https://www.netiq.com/documentation/netiqaccessmanager4/identityserverhelp/data/bookinfo.html).

To configure the user identification method for CloudAccess:

1. In the Access Manager Administration Console, click **Devices** > **Identity Servers** > **Edit** > **SAML 2.0** > **[Identity Provider]** > **User Identification**.
2. Click the **Edit** icon next to **Attribute Matching Settings**, select the attribute expression you defined in "Creating an Attribute Matching Expression to Use for the Identity Provider User Identification" on page 6, then click **OK**.
3. (Optional) If you have protected resources that require a user's name and password, configure a password retrieval method for the identity provider.

   In the **Post Authentication methods** pane, use the arrow keys to move the **Password Retrieval** method from the **Available Methods** list to the **Selected Methods** list.



User Identification Methods

☑ Attribute matching

☐ Prompt for password on successful match

Provisioning settings (undefined)

Attribute Matching settings (defined)

Post Authentication methods:

Selected methods:

(Note: Select the methods which don't require user identification)

Password Retrieval

Available Methods:

AttachmateGroup
Name/Password - Basic
Name/Password - Form
Secure Name/Password - Basic

☐ Logout on method execution failure

☑ Allow IDP to set session timeout

☐ Overwrite Temporary User

☐ Overwrite Real User

Assertion Validity Window 300 seconds

4. Select **Allow IDP to set session timeout**.
5. Click **OK** twice.
6. Update the Access Manager Identity Server.
7. Continue with "Configuring Attributes for the Identity Provider" on page 16.

Configuring Attributes for the Identity Provider

You must specify the attributes that CloudAccess can use to match the user to an account in the Access Manager user store. An authentication request and response contain these attributes.

1. In the Access Manager Administration Console, click **Devices > Identity Servers > Edit > SAML 2.0 > [Identity Provider] > Attributes**.
2. Select the attribute set that you created in "Creating an Attribute Set to Use for the Identity Provider Attributes" on page 5.
3. Select attributes from the **Available** list, and move them to the left side of the page.

   The attributes that you move to the left side of the page are the attributes you want to use for user matching.
4. Click **OK** twice.
5. Update the Access Manager Identity Server.
6. Continue with "Assigning the External Authentication Contract to Protected Resources" on page 16.

Assigning the External Authentication Contract to Protected Resources

You can use CloudAccess as the identity provider for back-end resources protected by Access Gateway. To do this, use the external authentication contract that you created for CloudAccess as the definition of how users authenticate to the protected resources.

For more information about configuring Access Gateway to protect resources, see "Configuring Protected Resources" (https://www.netiq.com/documentation/netiqaccessmanager4/accessgatewayhelp/data/prlist.html) in the *NetIQ Access Manager 4.0 Access Gateway Guide* (https://www.netiq.com/documentation/netiqaccessmanager4/accessgatewayhelp/data/bookinfo.html).

To assign the contract for the trusted identity provider to a protected resource:

1. Before you begin, ensure that you have configured a trusted relationship between the Access Manager Identity Server and the Access Gateway.

   This allows the authentication contracts that you created on the Identity Server to be available to Access Gateway.

   For information about setting up these Access Manager components to trust the Access Manager certificate authority, see "Using Access Manager Certificates" (https://www.netiq.com/documentation/netiqaccessmanager4/basicconfig/data/bfgfinm.html) in the *NetIQ Access Manager 4.0 Setup Guide* (https://www.netiq.com/documentation/netiqaccessmanager4/basicconfig/data/bookinfo.html).
2. In the Access Manager Administration Console, click **Devices** > **Access Gateways** > **Edit** > **[Reverse Proxy Name]** > **[Proxy Service Name]** > **Protected Resources**.
3. Select the **Resource View** of the Protected Resource list.
4. On the Resource View page, select the name of an existing protected resource, or click **New** and specify the display name for a new protected resource.
5. Configure the resource to use the external authentication contract that you created in "Creating an External Authentication Contract for the Identity Provider" on page 7.

   If no contracts are available, you have not configured a trusted relationship between the Access Gateway and the Identity Server.
6. Click **OK** to save the changes.
7. On the Protected Resources page, select the Resource View.

8. In the **Protected Resource** list, locate the resource you created, then ensure that it is enabled (displays a green check mark icon in the **Enabled** column).
9. Update the Access Gateway to apply the changes.
10. Continue with "Configuring Identity Provider-Initiated Logins" on page 17.

Configuring Identity Provider-Initiated Logins

You must complete some additional configuration steps to enable users to log in to the web service while also authenticating to the identity source. A login initiated by the identity provider allows users to start the login process at the CloudAccess appliance. This includes login using MobileAccess if it is enabled and configured on the appliance.

1. The user accesses the IdP-initiated login URL you provide.

   https://*appliance_DNS*/

2. The login page displays different appmarks for each application configured to work with the appliance.
3. The user clicks the appmark to access the web service.
4. CloudAccess redirects the login to the web service.
5. The user is authenticated to both the identity source and the web service at this point.

You must provide a link to the IdP-initiated login URL for users to access.

https://*appliance_DNS*/

You can also copy the auto-generated URL on each icon to provide as a link for the user.

Troubleshooting the CloudAccess and Access Manager Configuration

- Cannot Log In to CloudAccess or Cannot Access the Access Manager Appmark (page 17)
- MobileAccess Authentication Fails Across Multiple Tabs in the Same Browser (page 17)
- Password Retrieval Redirects the User to /nidp/app Instead of the Intended Protected Resource (page 18)

Cannot Log In to CloudAccess or Cannot Access the Access Manager Appmark

If you configured your system correctly, you should be able to test the federation by logging into CloudAccess from your desktop and selecting the Access Manager appmark. If you cannot access the protected resource, check for these common configuration problems:

- Missing or incorrect certificate in IdP trust store
- Contract is not satisfiable by an external identity provider
- Incorrect attribute mapping

You can also use the SAML tracer Firefox extension and the IdP logs to identify other issues.

MobileAccess Authentication Fails Across Multiple Tabs in the Same Browser

When you log in to MobileAccess and try to access a NetiQ Access Manager resource, you get this error:

```
Error: An Identity Provider response was received that failed to authenticate this
session. (300101013 A08BEB1E8489644E)
```

When you are logged in to MobileAccess and go to the Access Manager login page in the same web browser, Access Manager does not recognize you as already logged in.

For troubleshooting information, see *Troubleshooting cheat sheet - how to troubleshoot Access Manager 3.2 SAML issues (Technical Information Document 7014298)* (http://www.novell.com/support/kb/doc.php?id=7014298).

Password Retrieval Redirects the User to /nidp/app Instead of the Intended Protected Resource

When password retrieval is used to access a protected resource, Access Manager redirects the user to the `../nidp/app` and not to the Access Gateway protected resource that was originally defined in the target URL.

A workaround is available. Contact Technical Support if this is an issue in your deployment. A fix will be available in NetIQ Access Manager 4.0 SP1.