NetIQ®

NetIQ® CloudAccess

Connectors Guide

September 2014

www.netiq.com/documentation

Legal Notice

Contents

About this Book and the Library

The *NetIQ® CloudAccess Connectors Guide* provides installation and configuration information for 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

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

Driving your success is our passion

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

Our Solutions

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

Contacting Sales Support

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

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

Contacting Technical Support

For specific product issues, contact our Technical Support team.

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

Contacting Documentation Support

Our goal is to provide documentation that meets your needs. The documentation for this product is available on the NetIQ 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 13
- Section 1.2, "Connectors for Federated Single Sign-On and Provisioning," on page 18
- Section 1.3, "Connectors for Federated Single Sign-On," on page 18
- Section 1.4, "Connectors for Basic Single Sign-On," on page 19
- Section 1.5, "Connector for OAuth 2.0 Single Sign-On," on page 20
- Section 1.6, "Connector for Simple Proxy Single Sign-On," on page 20
- Section 1.7, "Connector for Bookmarks," on page 20
- Section 1.8, "Custom Connectors," on page 20
- Section 1.9, "License Information for Connectors," on page 21

1.1 Understanding Single Sign-On Methods

CloudAccess supports single-sign 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 13
- Section 1.1.2, "Basic Single Sign-On," on page 14
- Section 1.1.3, "OAuth 2.0 Single Sign-On," on page 17
- Section 1.1.4, "Simple Proxy Single Sign-On," on page 17
- Section 1.1.5, "Bookmarks," on page 18

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 vender-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. The user provides 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 the user initiates 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 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 application service provider.
6. The service provider validates the assertion and allows the user to access the application.

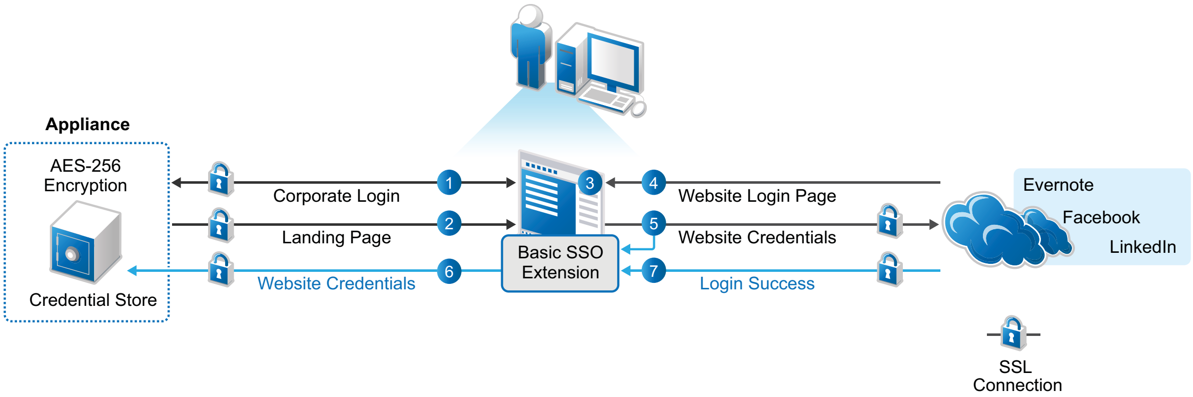
1.1.2 Basic Single Sign-On

Basic single sign-on provides an internal credentials store where users can save their credentials for third-party websites that require a password 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 the user's credentials for the site.

Basic SSO connectors work with the Basic SSO extension for the Chrome browser running on the user's computer to securely collect, store, retrieve, and replay the user credentials for a destination website. Users must log in to the website once in order for the extension to capture and store the credentials in the CloudAccess credential store. The user can choose whether to store the credentials for each destination website. If the user does not allow credentials to be saved for a website, the user must enter the site's credentials for each session.

Figure 1-1 depicts the user experience when the user clicks the appmark for a Basic SSO application.

Figure 1-1 *User's First-Time Login to the Website with Basic SSO*



The following describes the experience for Basic SSO the first time the user accesses the app:

1. In a Chrome browser, the user logs in to the CloudAccess login page using their corporate credentials.
2. The user sees the available applications on the landing page.
3. The user clicks the appropriate application icon.

   If the Basic SSO extension for the Chrome browser is not installed on the computer:

   a. The connector prompts the user to install the Basic SSO extension.

   b. The user accepts the prompt, and the appliance opens the Google Play Store in a new tab.

   c. The user installs the Basic SSO extension, then closes the Google Play Store tab to continue.

   d. The user returns to the landing page and clicks the appropriate application icon again.
4. A new tab opens for the login page of the application.
5. The user enters their user name and password for the destination website.

   The user must enter this separate user name and password once.
6. The extension asks if the user wants the credentials to be saved by CloudAccess, and the user allows the credentials to be saved.

   a. The extension captures the user name and password, and sends them to CloudAccess over an SSL connection.

      The extension obfuscates the user name and password with Base64 encoding before transmission.

   b. CloudAccess encrypts the site-specific credentials with AES-256 encryption, and then stores the encrypted data in the credential store that is part of the appliance.

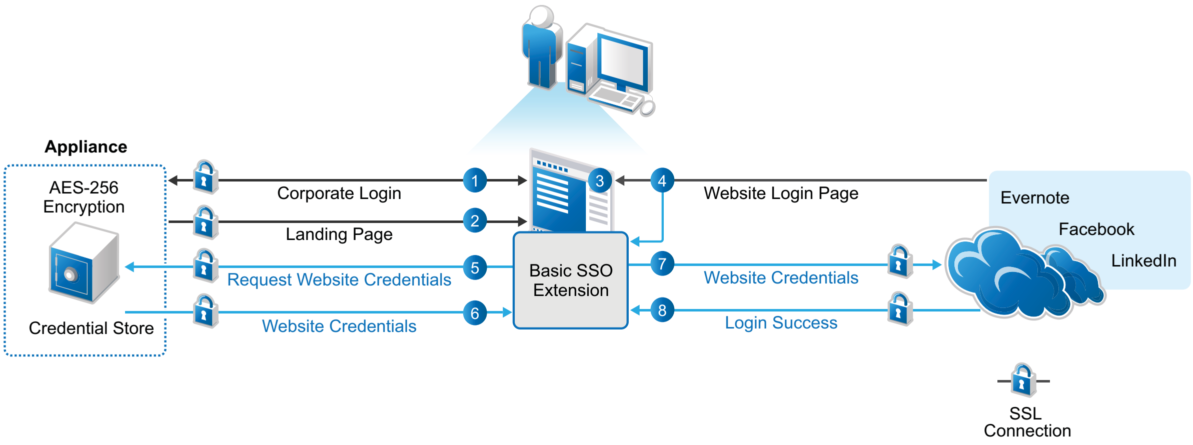
      The appliance encrypts the user name and password with an encryption key that is unique per user.
7. The website returns a success or failure indicator for the login.

   If the login succeeds, the browser opens to the application's website over an SSL connection.

   If the login fails, the browser returns the user to the website's login page to try again, and the extension requests that CloudAccess remove the saved credentials.

After the user allows the password to be stored securely, the user experiences single-sign-on access to the application in subsequent sessions. Figure 1-2 depicts the user experience when the user clicks the appmark for a Basic SSO application and the user's credentials are available in the credentials store.

Figure 1-2 *User's Single Sign-On Access to a Website with Basic SSO*



The following describes the experience for Basic SSO after the user stores credentials:

1. The user logs in to the CloudAccess login page using their corporate credentials.
2. The user sees the available applications on the landing page.
3. The user clicks the appropriate application icon.
4. A new tab opens for the login page of the application.
5. The Basic SSO extension requests that CloudAccess retrieve the user's user name and password for the site from the credential store.
6. CloudAccess retrieves the site-specific encrypted credentials from the credential store, decrypts them, and then sends the user name and password to the application's website over an SSL connection.

   CloudAccess obfuscates the user name and password with Base64 encoding before transmission.
7. CloudAccess logs in the user to the application's website. To the user, it appears as a single sign-on experience.

   If the user changes their login credentials for the destination website, the user will be prompted to log in again and the new credentials will be stored using the same process as for the initial setup.
8. The website returns a success indicator for the login, and the browser opens to the application's website over an SSL connection.

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, the user 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 their 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. The user provides 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 Simple Proxy Single Sign-On

Simple proxy single sign-on provides reverse proxy access to your enterprise web service through CloudAccess. If the web service requires user identity information to control access or content, you can configure the connector to inject the authenticated user's identity attributes in query strings and HTTP headers sent to the web service. However, the connector cannot be used to provide single sign-on for web services that require passwords for access. This proxy solution cannot inject the password. It does not support site redirects.

The following describes the experience for simple proxy access to a web service through CloudAccess:

1. The user provides 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 application.
5. (Conditional) CloudAccess sends identity information about the user in query strings and headers.
6. The website validates the resource request.
7. The user gains access to the resource.

1.1.5 Bookmarks

In CloudAccess, you can create bookmarks to 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. The user provides 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 three 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.

- "Connector for Google Apps for Business (SAML 2.0)"
- "Connector for Microsoft Office 365 (SAML 2.0 or WS-Federation)"
- "Connector for Salesforce (SAML 2.0)"

After you initialize the appliance, the connectors for Google Apps and Salesforce are automatically visible in the Applications palette on the Admin page of 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 in order 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. The connectors support either the SAML 2.0 protocol or the WS-Federation protocol.

The connector for NetIQ Access Manager ships with the appliance. You can use this connector if you have a CloudAccess license or a MobileAccess-only license as well as an account for Access Manager. For more information, see NetIQ Access Manager (SAML 2.0).

You can download additional connectors from NetIQ Downloads (https://dl.netiq.com/). For configuration information, see the following:

- "Connector for Accellion (SAML 2.0)"
- "Connector for ADFS (SAML 2.0)"
- "Connector for ADFS (WS-Federation)"
- "Connector for Azure (WS-Federation)"
- "Connector for Box (SAML 2.0)"
- "Connector for Jive (SAML 2.0)"
- "Connector for ServiceNow (SAML 2.0)"
- "Connector for VMware vCloud (SAML 2.0)"
- "Connector for WebEx (SAML 2.0)"
- "Connector for Zoho (SAML 2.0)"

You can also create custom connectors for federated single sign-on and logout by using the NetIQ Access Connector Toolkit. For more information, see "Creating Custom Connectors".

After you download a 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 (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, Linked In, and Facebook. Basic SSO connectors work with the Basic SSO extension for the Chrome browser 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.

You can download the connectors for Basic SSO from NetIQ Downloads (https://dl.netiq.com/). After you download a connector, you must import it to CloudAccess to make it available in the **Applications** palette in the CloudAccess administration console. For more information, see "Connectors for Basic SSO".

You can also create custom connectors for Basic SSO by using the NetIQ Access Connector Toolkit. For more information, see "Creating Custom Connectors".

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 "Connector for OAuth2 Resources".

1.6 Connector for Simple Proxy Single Sign-On

CloudAccess provides a connector for Simple Proxy that gives users reverse proxy access to your enterprise web service through CloudAccess. The connector ships with the appliance.

CloudAccess supports using multiple instances of the connector for Simple Proxy. Each instance points to a different destination website path. You can use this connector if you have a CloudAccess license or a MobileAccess-only license as well as access to the destination web path.

For more information, see "Connector for Simple Proxy".

1.7 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 or a MobileAccess-only license as well as access to the destination web service.

For more information, see "Connector for BookMarks".

1.8 Custom Connectors

CloudAccess provides the NetIQ 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 NetIQ Technical Support (NTS) (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 NetIQ 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 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 "Creating Custom Connectors".

1.9 License Information for Connectors

A CloudAccess license entitles you to use any of the connectors mentioned in this guide, including custom connectors.

A MobileAccess-only license entitles you to use only the following three connectors on the **Applications** palette in the CloudAccess administration console. All other connectors, including custom connectors, are CloudAccess-only features and require a CloudAccess license.

- "Connector for NetIQ Access Manager (SAML 2.0)"
- "Connector for Bookmarks"
- "Connector for Simple Proxy"

For more information, see "Understanding Product Licensing".

2 Configuring Connectors

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

2.1 Requirements for Connectors

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

❒ Ensure that the **Display Name** for each configured instance of a connector is 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 will 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 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

After you configure the connectors for applications and set entitlements for users, you must provide a way for users to access the applications. CloudAccess provides the following portal pages for the users:

- **Login page:** The login page allows users to enter their corporate credentials for authentication, such as user name and password. CloudAccess authenticates a user against your identity sources.

  The login page also exposes the authentication extensions of the appliance, such as Google reCAPTCHA.

  The URI for the login page is the public DNS name of the appliance. Provide this URl to your users.

  `https://`*`appliance_dns_name`*

- **Authentication code page:** The authentication code page requires users to enter a one-time password if a one-time password extension is enabled for one or more applications.

  If you enable the same extension for all applications, users see the authentication code page immediately after logging in. Otherwise, they see it when they click any one of the enabled applications.

  For example, you can enable an extension for the Time-Based One-Time Password with Google Authenticator or the One-Time Password with the OATH OTP or Smartphone option for NetIQ Advanced Authentication.

- **Landing page:** The landing page contains the appmarks (linked icons) for accessing the applications that a user is entitled to use. The landing page appears after a user enters valid credentials and responds successfully to any additional authentication prompts.

  This page displays appmarks for an application only after the related connector is configured properly and if the user has an entitlement for the application. When an authenticated user clicks an appmark, CloudAccess shares identity information about the user with the application in order to establish the user’s session.

  In a desktop browser, users can personalize the Card Style used to display appmarks. Right-click the appmark to view or copy the Login URL. Mouse over the appmark to view the application name.

  For more information, see Section 2.5, “Configuring Appmarks for Connectors,” on page 29.

2.4 How CloudAccess Provisions User Accounts

The connector for Google Apps for Business, the connector for Salesforce, and the connector for Microsoft Office 365 support both account provisioning and single sign-on. These three 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 25
- Section 2.4.2, "Understanding Provisioning," on page 25
- Section 2.4.3, "Samples of Account Creations," on page 26
- Section 2.4.4, "CloudAccess Naming Convention for Newly Provisioned Accounts," on page 27
- Section 2.4.5, "Matching Criteria for Merging Existing Accounts," on page 28

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. You must map authorizations for the appropriate roles (groups) to enable their entitlements to the applications. Users must log in with a corporate identity in order to access their provisioned account.

The connectors cannot provision accounts for users in a Self-Service User Store (SSUS) identity source or a SAML 2.0 Inbound (SAML2 In) internal identity store.

IMPORTANT: Although CloudAccess does not prevent it, do not create policy mappings between provisioning applications and SSUS or SAML2 In 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 two 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:** You can configure the SaaS connectors to allow users control of when their accounts are provisioned. A configuration option is available on the connector for Google Apps and the connector for Salesforce 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 SaaS 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 28.
- If no match is found, Cloud creates a new account. For more information, see Section 2.4.4, “CloudAccess Naming Convention for Newly Provisioned Accounts,” on page 27.

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 or WS-Federation. CloudAccess sends information about an authenticated user to the destination application by 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 24.

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** is enabled).

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 27.
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.
8. CloudAccess uses single sign-on to authenticate the users to the SaaS application.

Sample experience with user-controlled account provisioning:

1. The administrator selects the **Prompt users for an existing account before provisioning** option when configuring the connector for the SaaS application.
2. The administrator maps one or more SaaS 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 SaaS applications they are entitled to access.
6. Users click the appmark for the entitled SaaS application.
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. Users select **I do not have an existing account. Create one for me.**

   or

   Users select **I already have an existing account. These are my credentials**, then specify their credentials for the existing SaaS account.
9. Regardless of the option the user selects, CloudAccess searches for an existing, matching account.
10. 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 27.
11. CloudAccess uses single sign-on to authenticate the users to the SaaS application.

2.4.4 CloudAccess Naming Convention for Newly Provisioned Accounts

CloudAccess contains a defined 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
Active Directory	sAMAccountName	sAMAccountName@*Federated Domain Name*	sAMAccountName@*Salesforce Domain Name*
eDirectory	CN	CN@*Federated Domain Name*	CN@*Salesforce Domain Name*
JDBC	CN	CN@*Federated Domain Name*	CN@*Salesforce Domain Name*

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 28
- "Matching Criteria for the Connector for Office 365" on page 28
- "Matching Criteria for the Connector for Salesforce" on page 29

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 in 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	
Active Directory		objectGUID
		employeeID
eDirectory		GUID
		workforceID
JDBC		
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		

2.5 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 on the landing page that they are entitled to see, according to the application settings for public access or policy mappings for the application to identity source roles (groups).

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 mobile device before the appmark appears. To do a refresh, the user does the standard "pull-to-refresh" action on the Applications page in the MobileAccess app. (This action is used in mail and other common applications on the mobile device.)

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.

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

- Section 2.5.1, "Understanding Appmark Options," on page 30
- Section 2.5.2, "Mobile Device Workflow using Safari or Chrome," on page 32
- Section 2.5.3, "Mobile Device Workflow with Internal Viewer," on page 32
- Section 2.5.4, "Mobile Device Workflow from Bookmarks," on page 32
- Section 2.5.5, "Configuring an Appmark for the Desktop Browser or Mobile Device," on page 33
- Section 2.5.6, "Creating Multiple Appmarks for an Application," on page 34
- Section 2.5.7, "Using Appmark Variables," on page 34
- Section 2.5.8, "Policy Mapping for Non-Public Appmarks," on page 34

2.5.1 Understanding Appmark Options

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 2.5.6, "Creating Multiple Appmarks for an Application," on page 34.

Public

This option is available only for appmarks configured for Simple Proxy, Bookmark, 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, 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 2.5.7, “Using Appmark Variables,” on page 34.

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 their usability for users.

iOS devices

Enables the appmark to be visible on supported iOS mobile devices in the MobileAccess app on the Applications page.

Android devices

Enables the appmark to be visible on supported Android mobile devices in the MobileAccess app on the Applications page.

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 MobileAccess 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 MobileAccess app launches Chrome and directs it to the application. If Chrome is not installed on the mobile device, the user is taken to the App Store to install it.
- **Internal viewer:** When the user opens the MobileAccess app on the mobile device and taps the appmark, the MobileAccess app opens an embedded HTML viewer and directs it to the application. This view is similar to the Safari and Chrome options, except that the user does not have to leave the MobileAccess window. The application opens within the MobileAccess app window, and the user can tap the app name (as defined by the administrator when configuring 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.

2.5.2 Mobile Device Workflow using Safari or Chrome

When you select **Safari** or **Chrome** from the appmark **Launch with** list, MobileAccess opens the application in a new tab in the browser by using the MobileAccess proxy.

The browser workflow on the mobile device is as follows:

1. The end user opens the MobileAccess app on the mobile device.
2. (Conditional) If it is configured, the user is prompted for and enters an application PIN.
3. The user sees a list of protected resources and selects a protected resource.
4. The MobileAccess app starts Safari or Chrome and directs it to the protected resource via the MobileAccess proxy by opening a new tab in the browser.
5. The end user is allowed access to the protected resource.
6. In Google Chrome, the user can tap the button in the top left of the navigation bar to close the current tab and return to the MobileAccess app.

2.5.3 Mobile Device Workflow with Internal Viewer

When you select **Internal viewer** from the appmark **Launch with** list, MobileAccess opens an embedded HTML viewer and directs it to the protected resource by using the MobileAccess proxy.

The workflow on the mobile device is as follows:

1. The end user opens the MobileAccess app on the mobile device.
2. (Conditional) If it is configured, the user is prompted for and enters an application PIN.
3. The user sees a list of protected resources and selects a protected resource.
4. The MobileAccess app opens an embedded HTML viewer and directs it to the protected resource using the MobileAccess proxy.
5. The end user is allowed access to the protected resource.

2.5.4 Mobile Device Workflow from Bookmarks

When a user opens a protected bookmarked application in a Safari browser, MobileAccess prompts the user for the application PIN, then allows the user to access the bookmarked application.

The workflow using bookmarks on the mobile device is as follows:

1. The end user opens Safari on the mobile device.
2. The end user selects a bookmark that points to a URL protected by MobileAccess (i.e., a protected resource).

3. The end user is redirected to the MobileAccess app.
4. (Conditional) If it is configured, the user is prompted for and enters an application PIN.
5. The end user is redirected back to Safari and the bookmarked URL (protected resource).
6. The end user is seamlessly allowed access to that bookmarked application.

2.5.5 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 2.5.1, "Understanding Appmark Options," on page 30.

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.

2.5.6 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 2.5.7, “Using Appmark Variables,” on page 34.
5. Specify whether the application should be accessible from a desktop browser or a mobile device, or both, and complete the appropriate fields. For more information about available options, see Section 2.5.1, “Understanding Appmark Options,” on page 30.
6. Click **OK**, then click **Apply** to update the appliance.

2.5.7 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.

2.5.8 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.

2.6 Downloading and Importing Single Sign-On Connectors

NetIQ provides additional connectors for federated single sign-on and basic single-sign-on. The connectors are available for download from NetIQ Downloads. You must import the connector into CloudAccess in order to use it.

To download and import a connector template to CloudAccess:

1 Download the template file for the desired connector:

  1a Launch a web browser on the computer that you use to manage CloudAccess.

  1b Go to NetIQ Downloads (https://dl.netiq.com/).

  1c Search for the desired connector, such as the connector for WebEx.

  1d Click the connector's download link and follow the instructions to download the file.

  1e When you are prompted, log in with your NetIQ customer account credentials, then continue with the download.

  1f Extract the connector's ZIP file and release notes to a local directory.

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

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

3 Click the **Tools** icon on the toolbar, then click **Import connector template**.

4 Click **Browse,** then browse to and select the ZIP file that you extracted from the connector's download file.

5 Click **Import**.

  The **Applications** palette displays the connector that you imported.

6 Proceed to the configuration instructions for the desired connector.

After you import the connector, you must configure the connector settings in CloudAccess. For more information, see the requirements and configuration information for the desired connector.

3 Creating Custom Connectors

CloudAccess provides the NetIQ 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 NetIQ Technical Support (NTS) (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 NetIQ 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.

3.1 Accessing the Access Connector Toolkit

The Access Connector Toolkit is a web application that you access through the CloudAccess appliance. It requires administrator credentials.

To access the toolkit:

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

   `https://`*`appliance_dns_name`*`/css/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.

- Section 3.2.1, “License Information,” on page 38
- Section 3.2.2, “Toolkit Compatibility,” on page 38
- Section 3.2.3, “Provisioning Support,” on page 38

3.2.1 License Information

The Access Connector Toolkit is a CloudAccess-only feature. If you purchased MobileAccess without CloudAccess, your license entitles you to a 90-day trial of CloudAccess, including the Access Connector Toolkit. At the end of the trial period, you are expected to purchase the appropriate license for CloudAccess or discontinue use of the CloudAccess-only features. For more information, see “Understanding Product Licensing” in the *NetIQ® CloudAccess and MobileAccess Installation and Configuration Guide*.

3.2.2 Toolkit Compatibility

The Access Connector Toolkit contains new functionality for the CloudAccess 2.1 release. In order 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 into a toolkit that came with a prior version of CloudAccess. The import fails.

3.2.3 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 25.

3.3 Federation Requirements for the Application Service Provider

As you explore the features of the Access Connector Toolkit, refer to the definitions in this section to understand the type of information you will need to collect from the destination web service or application.

assertion

A SAML 2.0 assertion is a package of identity attributes for an authenticated user that is sent from the trusted identity provider to the service provider.

assertion properties

The properties of the assertion include the following information:

- The recipient of the assertion.
- The LDAP identity attribute to use when federating users with the destination application service provider. Does the NameID require an email address format, or does it require unspecified format?
- The URL where CloudAccess should redirect the end user's session after the user logs in successfully with the URL provided on the connector configuration page.
- The binding method to use for identity information sent to the destination provider. For SAML 2.0, the only supported binding method is POST.

assertion attributes

The provider should provide a technical document that describes the attributes that are required for an assertion, such as the user's name or email address. It can include the attributes that are required to assign roles. The SAML assertion typically requires the nameID attribute. You must map the SAML assertion attributes to the matching attributes in your identity source.

entityID

The entityID is a field from the metadata that uniquely identifies that particular service provider, such as *sp_domain_name*.

For example:

google.com

The entity ID might use information from the federation instructions, or from a setting completed on the Configuration page when you deploy the connector.

federation instructions

The federation instructions 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 will 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.

It provides the following information about your appliance if the login is initiated by the service-provider, such for connectors that use the WS-Federation protocol:

- The WS-Federation Passive URL
- The X.509 signing certificate for the appliance

metadata

The metadata is the configuration information that the application service provider uses to establish communications with the identity provider in an federation trust relationship. This usually includes a login URL or a customer-specific domain name, which is called the Assertion Consumer Service URL. Service providers allow you to export the required metadata to an XML file, or they provide the metadata in a public URL. The auto-generated metadata file from the service provider will not work as is. You must manually change the values to match your actual deployment environment.

The metadata usually includes the following information:

- The entityID for the service provider.
- The URL that receives the user identity information.
  - For SAML 2.0, the Assertion Consumer Service URL is where the assertion is posted by the browser. For example:

    `https://www.google.com/a/${`*`customer-domain`*`}/acs`
  - For WS-Federation, the Login URL is where the security token is posted by the browser. It corresponds to the `PassiveRequestorEndpoint` field from the metadata.
  - For SAML2 In, the Single Sign-on Service URL is where the AuthnRequest will be posted. It corresponds to the `SingleSignOnService` field with a Post binding from the metadata.

    `https://accessmanager.base.url/nidp/saml2/sso`
- The logout URL corresponds to the `SingleLogoutService` field from the metadata.
- The logout URL Binding (HTTP Post or Redirect)

  The logout response URL
- The X.509 signing certificate

protocol binding

The protocol binding is the method used for transmitting assertions between the authenticating identity provider and the service provider. CloudAccess supports the Redirect and Post bindings for service-provider-initiated SSO, and the Post binding for identity-provider-initiated SSO.

nameID

The nameID is the attribute in the identity source that uniquely identifies the user. You must know whether this attribute requires the email address format or an unspecified format.

new settings

The new settings are appliance-specific settings that you want to allow the administrators to set when they configure the connector for an appliance.

For example:

- Customer-specific sections of the Assertion Consumer Service URL
- Connector-specific setting, such as a customer domain

security token

A WS-Federation security token is a package of identity attributes for an authenticated user that is sent from the trusted identity provider to the service provider. The provider should provide a technical document that describes the attributes that are required for the token, such as the user's name or email address. It can include the attributes that are required to assign roles.

signing certificate

The signing certificate is the X.509 certificate that identifies CloudAccess to the service provider. If you specify that the certificate is required by the service provider, the template automatically retrieves the appliance's certificate and inserts it in the Federation Instructions when you deploy the connector. You use the certificate when you set up the federated single sign-on for the application.

template properties

The template properties define the following information for the connector:

- Type of connector and type name (based on the template wizard)
- The unique name for the template file (target name)
- A brief description used as the connector name
- 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.

3.4 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 3.4.1, "SAML 2.0 Requirements for the Application Service Provider," on page 41
- Section 3.4.2, "Planning for a SAML 2.0 Connector," on page 42
- Section 3.4.3, "Creating a SAML 2.0 Connector Template for an Application," on page 42

3.4.1 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.

3.4.2 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. For more information, see Section 3.3, "Federation Requirements for the Application Service Provider," on page 39.

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".

3.4.3 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, 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, the 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 3.2.3, "Provisioning Support," on page 38.

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

8 Proceed to Section 3.9, "Exporting a Connector Template," on page 50 to finish creating the new connector.

3.5 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 3.5.1, "WS-Federation Requirements for the Application Service Provider," on page 43
- Section 3.5.2, "Planning for a WS-Federation Connector," on page 43
- Section 3.5.3, "Creating a WS-Federation Connector Template for an Application," on page 44

3.5.1 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 the customer to enable and configure WS-Federation SSO

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

3.5.2 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. For more information, see Section 3.3, "Federation Requirements for the Application Service Provider," on page 39.

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".

3.5.3 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, 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, the 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 3.2.3, "Provisioning Support," on page 38.
7. Click **Save** to save the new connector template.
8. Proceed to Section 3.9, "Exporting a Connector Template," on page 50 to finish creating the new connector.

3.6 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. 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 in order 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 3.6.1, "Understanding SAML2 In Identity Sources," on page 45
- Section 3.6.2, "Requirements for Using SAML2 In Identity Sources," on page 46
- Section 3.6.3, "SAML2 In Requirements for the Identity Provider," on page 47
- Section 3.6.4, "Planning for a SAML2 In Connector," on page 47
- Section 3.6.5, "Creating a SAML2 In Connector for an Identity Provider," on page 48

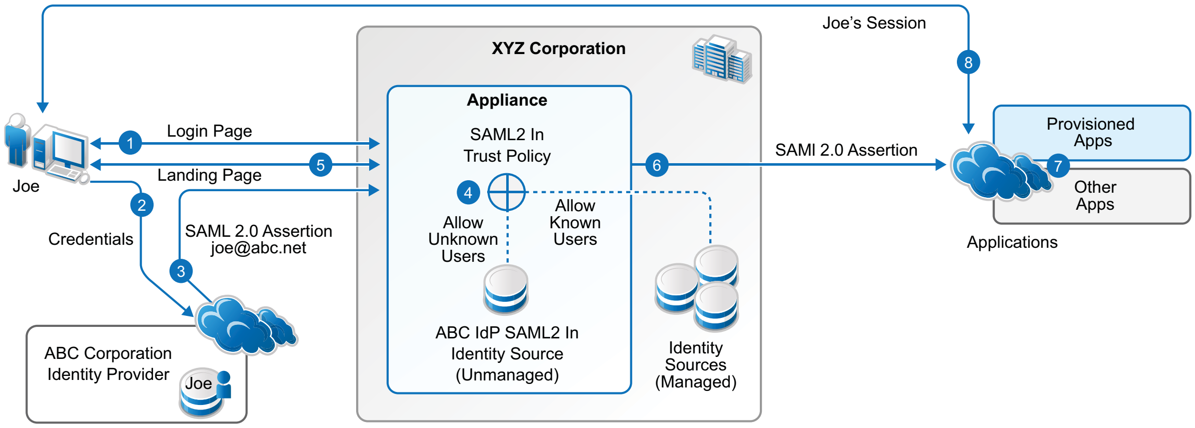
3.6.1 Understanding SAML2 In Identity Sources

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. 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 SAML 2 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 SAML 2 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 their logged-in identity.
8. The application service provider establishes a session with the user.

3.6.2 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 their identity provider account, the user 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 25.

❐ Users in a SAML2 In internal identity store are not supported in administration roles for the appliance.

3.6.3 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 3.10, "Importing and Configuring Custom Connectors," on page 50.

❒ 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 in order to uniquely identify the user. For more information about the role of email addresses for SAML2 In users, see Section 3.6.2, "Requirements for Using SAML2 In Identity Sources," on page 46.

3.6.4 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".

3.6.5 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 3.9, "Exporting a Connector Template," on page 50 to finish creating the new connector.

3.7 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 3.7.1, "Basic SSO Requirements," on page 48
- Section 3.7.2, "Planning for Basic SSO," on page 49
- Section 3.7.3, "Creating a Basic SSO Connector Template for a Web Service," on page 49

3.7.1 Basic SSO Requirements

Gather the following information to create your 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 a Chrome web browser running on a desktop or laptop computer. It does not support access from mobile devices.
- ❐ The NetIQ Basic SSO extension is compatible only with the Chrome web browser. A user must install the extension in the Chrome browser one time on each desktop or laptop they use to access the Basic SSO websites. The extension is available for free from the Google Play Store.

3.7.2 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)?
- What is the criteria for a successful login or a failed login?

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

3.7.3 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 an administrator to the Access Connector 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)
   - A brief description used as the connector name
   - 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. Under **Target Domain**, click **New**, then define the domain for the desired application for Basic SSO authentications.

   Domain: Specify the URL of the application for Basic SSO authentications.

   Path ID: Specify the ID of the path for the desired action. For example login or success.

   Path: Specify the URL for each ID. For example the login URL, or the success URL.
5. On the **Form** tab, create the form for the Basic SSO connector.

   The **Form** tab allows you define the HTML form for the desired application. The toolkit lists 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.

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

7 Proceed to Section 3.9, "Exporting a Connector Template," on page 50 to finish creating the new connector.

3.8 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 **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 3.10, "Importing and Configuring Custom Connectors," on page 50.

3.9 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 3.10, "Importing and Configuring Custom Connectors," on page 50.

3.10 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 NetIQ Technical Support or NetIQ 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 3.10.1, “SAML2 and WS-Fed Custom Connectors,” on page 51
- Section 3.10.2, “SAML2 In Custom Connectors,” on page 52
- Section 3.10.3, “Basic SSO Custom Connectors,” on page 53

3.10.1 SAML2 and WS-Fed Custom Connectors

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

1. Copy the custom connector ZIP file to the computer where you administer CloudAccess.
2. Log in as an administrator to the CloudAccess administration console:

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

3. On the Admin page, click the **Tools** icon on the toolbar, then click **Import connector template**.
4. Browse to and select the custom connector ZIP file, then click **Import**.

   The connector appears in the **Applications** palette.
5. Drag and drop the new custom connector from the **Applications** palette to the **Applications** panel.

   The Configuration window opens automatically for the initial configuration. To view or reconfigure the settings later, click the connector icon, then click **Configure**.
6. 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.
7. 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.
8. Click the **Appmarks** tab, then review and edit the default settings for the appmark.

   For more information, see Section 2.5, “Configuring Appmarks for Connectors,” on page 29.
9. Click **OK** to save the configuration.
10. On the Admin page, click **Apply** to commit the changes to the appliance.
11. Wait until the configuration changes have been applied on each node of the CloudAccess cluster.
12. Log in to 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 7 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.
13. In the CloudAccess administration console, click **Policy** in 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 *NetIQ® CloudAccess and MobileAccess Installation and Configuration Guide*.

14 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
```

3.10.2 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 3.6.1, "Understanding SAML2 In Identity Sources," on page 45 and Section 3.6.2, "Requirements for Using SAML2 In Identity Sources," on page 46.

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

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

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

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

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

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

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

5 Drag and drop the new custom connector from the **Identity Sources** palette to the **Identity Sources** panel.

The Configuration window opens automatically for the initial configuration. To view or reconfigure the settings later, click the connector icon, then click **Configure**.

6 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.

7 Under **Assertion Attribute Mappings**, map the SAML Assertion attributes to the appropriate attributes in your identity source.

8 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 3.6.2, "Requirements for Using SAML2 In Identity Sources," on page 46.

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

For more information about unknown and known users, see Section 3.6.1, "Understanding SAML2 In Identity Sources," on page 45.

9 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.

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

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

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

13 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.

14 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 9 and the `SPSSODescriptor` from Step 13.

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.

---

15 (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** in 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 *NetIQ® CloudAccess and MobileAccess Installation and Configuration Guide*.

16 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.

3.10.3 Basic SSO Custom Connectors

To import and configure a custom connector for Basic SSO:

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

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

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

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

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

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

5 Drag and drop the new custom connector from the **Applications** palette to the **Applications** panel.

The Configuration window opens automatically for the initial configuration. To view or reconfigure the settings later, click the connector icon, then click **Configure**.

6 Connectors for Basic SSO do not require additional configuration.

7 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).

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 (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 *NetIQ® CloudAccess and MobileAccess Installation and Configuration Guide*.

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

```
https://appliance_dns_name
```

4 Connector for Google Apps for Business (SAML 2.0)

The connector for Google Apps for Business provides automated provisioning of accounts from the identity sources to Google Apps. The connector also provides federated single sign-on access to Google Apps with SAML 2.0 through CloudAccess. The connector allows CloudAccess to authenticate a user against your identity sources and to share this authentication with Google Apps in order 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 for Business:

4.1 Connector Requirements

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 25.

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

- ❐ A CloudAccess system, installed and configured
- ❐ A valid Google Apps for Business account
- ❐ Provisioning APIs enabled on the account
- ❐ An administrative account and password

4.2 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.

4.3 Configuring the Connector for Google Apps for Business

The connector for Google Apps provides user account provisioning and single sign-on access to Google Apps for Business domains. After users log in to CloudAccess, SAML authentication is used to automatically authenticate (single sign-on) users to Google Apps for Business. 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 and drop the SAML 2.0 connector for Google Apps for Business from the **Applications** palette to the **Applications** panel.

The Configuration window opens automatically for the initial configuration. To view or reconfigure the settings later, click the connector icon, then click **Configure**.

4 Provide a unique display name for the connector to appear on the Admin and landing pages, and also provide the administrator logon credentials and domain 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 at Google Apps for Business. Otherwise, you must manually configure the parameters at Google Apps for Business.

6 (Conditional) Select **Prompt users for an existing Google Apps account before provisioning** if you want to give users control of when their accounts are provisioned. For more information, see Section 4.2, "Understanding Google Apps Provisioning," on page 56.

7 (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.

8 (Conditional) If you did not select the **Automatically configure SSO settings** option, click **Federation Instructions**. Read the instructions provided to configure the connector for Google Apps for Business to allow single sign-on for users, then complete the following steps:

8a Copy and paste the text of the signing certificate provided in the **Federation Instructions** into a file, then save the file.

NOTE: Ensure that you use a text editor that does not introduce hard returns or additional white space. Otherwise, the certificate file may be improperly formatted and unusable. For example, use Notepad instead of Wordpad.

8b Log in to the Google Apps Dashboard with your administrator account.

8c Navigate to **Advanced Tools > Set up single sign-on (SSO)**.

8d Provide the following information:

Enable Single Sign-on: Select this option.

Sign-in page URL: Specify the value provided for the **Single Sign-on URL** in the **Federation Instructions**.

Sign-out page URL: Specify the value provided for the **Single Logout URL** in the **Federation Instructions**.

Change password URL: This is the page that the URL will redirect to when a user clicks **Change Password**. (This is not part of the federation per se, but Google requires a value for this field.)

Verification certificate: Upload the file into which you copied the signing certificate text above.

Use a domain specific issuer: Select this option. This changes the value sent in the SAML request to be `google.com/a/`*`google_apps_domain`* instead of google.com. For more information, see SSO (Single Sign-On) (http://support.google.com/a/bin/answer.py?hl=en&answer=60224) in the documentation for Google Apps for Business.

Network masks: Leave blank. This option is not applicable to SAML configuration.

8e Click **Save Changes**.

9 Click **OK**.

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

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

12 Perform policy mapping to specify entitlements for identity source groups.

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

13 (Optional) To provide users with access to Google Apps Mail from supported mobile devices, click the configured connector on the **Applications** panel, then click **Enable email proxy**.

14 (Optional) Modify default appmarks or configure new appmarks to specify how users should access the Google Apps applications.

15 Add users to the appropriate identity source group to trigger user account provisioning to Google Apps.

16 (Conditional) If required, grant approvals for mapped authorizations.

User accounts that have been provisioned to Google Apps for Business using CloudAccess must authenticate through CloudAccess. Direct logins to Google Apps for Business are not allowed. For more information, see the SAML SSO section of the Google Apps for Business website.

4.4 Configuring Appmarks for Google Apps

After you have configured the connector for Google Apps for Business, you can configure appmarks to specify how users should access the applications. By default, the connector includes three appmarks that are configured for the Calendar, Mail, and Drive applications. You can modify these default appmarks or create new ones.

To configure appmarks:

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 for Google Apps on the **Applications** panel, then click **Configure**.

3 Click the **Appmarks** tab.

4 Modify each appmark as needed. For more information about configuring appmarks, see Section 2.5, “Configuring Appmarks for Connectors,” on page 29.

5 Click **OK**.

6 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.

After the appliance has finished applying your changes, the appmarks appear on the landing page or the Applications page of the MobileAccess app for users to whom you have granted access.

NOTE: If you are upgrading your environment from CloudAccess 1.5 to CloudAccess 2.0, you must map new appmarks for any connectors for Google Apps that you configured in CloudAccess 1.5. For more information, see “Upgrading Your Environment” in the *NetIQ® CloudAccess and MobileAccess Installation and Configuration Guide*.

4.5 Configuring Multiple Connectors for Google Apps for Business

CloudAccess can support Google Apps domains by using multiple instances of the connector for Google Apps for Business. 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.)

5 Connector for Microsoft Office 365 (SAML 2.0 or WS-Federation)

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 in order 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 NetIQ 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. However, the connector does not appear automatically in the Applications palette of the Admin page. You must run the connector for Office 365 installer on your Windows Management Server in order to connect the web application to CloudAccess.

IMPORTANT: The connector for Office 365 is a CloudAccess-only feature and is not included in the MobileAccess-only license. For more information, see "Understanding Product Licensing" in the *NetIQ® CloudAccess and MobileAccess Installation and Configuration Guide*.

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

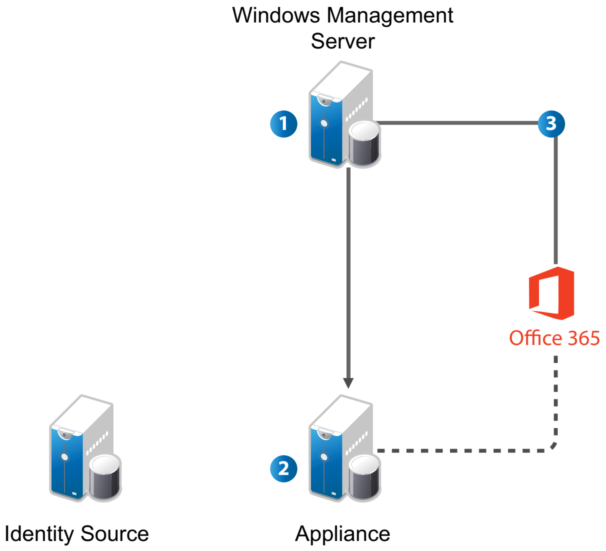
5.1 How the Connector for Office 365 Works

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

- Section 5.1.1, "Setup and Configuration," on page 60
- Section 5.1.2, "User Provisioning," on page 61
- Section 5.1.3, "User Login to Office 365," on page 61

5.1.1 Setup and Configuration

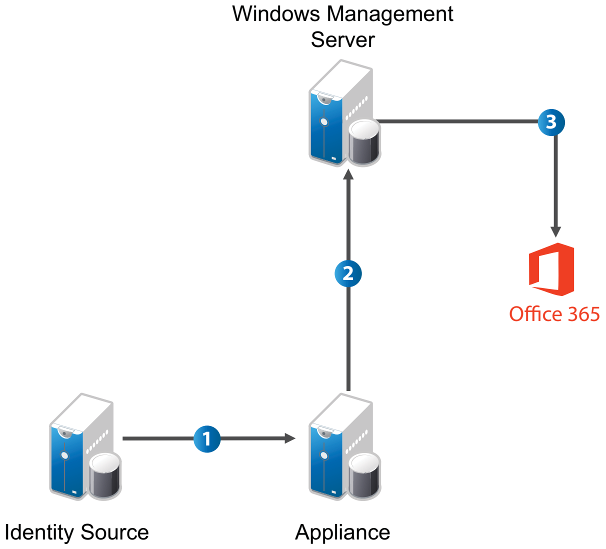
The following figure illustrates the basic setup and configuration steps.



1. Run the connector for Office 365 installer using the `.msi` file. For more information, see Section 5.3, "Installing the Connector for Office 365," on page 63.
2. Create a trust relationship between the CloudAccess appliance and Office 365.
3. Configure user provisioning to Office 365.

5.1.2 User Provisioning

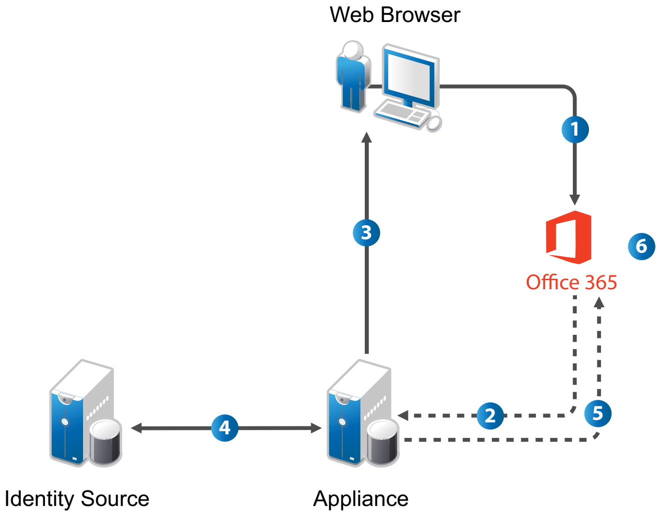
The following figure illustrates the workflow in provisioning users.



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.

5.1.3 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 sources. 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.

5.2 Connector Requirements

The connector for Office 365 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 25.

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.

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

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 a Windows Management Server on which to install the connector. 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 7 with the Web Server (IIS) role enabled and the ASP.NET 4.x feature added. 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/net/downloads) 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).

❐ 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. The 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. The 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*

5.3 Installing the Connector for Office 365

You must install the connector for Office 365 on a Windows Management Server.

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 domain name to Office 365.

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 a single unique Office 365 federated domain.

---

3 Verify that the Windows server where you plan to install the connector has the prerequisite software installed. For more information, see Section 5.2, "Connector Requirements," on page 62.

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

4a Create a self-signed certificate in IIS Manager.

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

4c Restart the IIS service.

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-1.5.1.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-1.5.1.msi /qb
AG4CHOSTNAME="DNS_of_appliance" AG4CADMIN="CloudAccess_Admin_username"
AG4CADMINPASS="CloudAccess_Admin_password"
O365ADMIN="Office365_Admin_username" O365ADMINPASS="Office365_Admin_password"
O365FEDDOMAIN="Office365_Federated_Domain_Name" O365USAGELOCATION="US"
```

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-1.5.1.msi /L*V "C:\log\example.log"
```

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

After you have installed the connector, when you return to the CloudAccess administration console, the connector for Office 365 icon is automatically moved to the Applications panel. No additional configuration is required, but you can configure appmarks for different Office 365 applications, and map users to the appropriate applications to set their entitlements. For more information about configuring appmarks, see Section 5.5, "Configuring Appmarks for Office 365 Applications," on page 65.

5.4 Validating the Connector for Office 365

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

1 Verify that the connector for Office 365 appears on the Applications panel of the Admin page of the console.

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

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

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

5 Specify the user name of `user@domain` where the domain is the federated domain name specified in Step 2 on page 63.

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**.

5.5 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 and/or in the MobileAccess app for entitled users. For more information, see Chapter 2.5, "Configuring Appmarks for Connectors," on page 29.

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".

To configure appmarks:

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 for Office 365 on the **Applications** panel, then click **Configure**.
3. Click the **Appmarks** tab.
4. Modify appmarks as needed.

   For more information about configuring appmarks, see Chapter 2.5, "Configuring Appmarks for Connectors," on page 29.

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

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

After the appliance has finished applying your changes, the appmarks appear on the landing page or in the MobileAccess app for users to whom you have granted access.

5.6 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.

5.7 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.

5.8 Upgrading the Connector from SAML 2.0 to WS-Federation

In CloudAccess 2.1, 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 in order 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** in order 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.

5.9 Uninstalling the Connector for Office 365

The connector for Office 365 consists of multiple components. To correctly uninstall the connector, log in to the Windows server as an administrator and use the uninstall function in Windows Control Panel.

Using the Control Panel to uninstall the connector deletes the connector from the CloudAccess Admin page. 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.

Alternatively, you can uninstall the connector by running the following command on the Windows server:

```
msiexec /x netiq-office365-connector-1.5.1.msi /qb
```

5.10 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.

6 Connector for Salesforce (SAML 2.0)

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 in order 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 NetIQ 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 6.1, "Connector Requirements," on page 67
- Section 6.2, "Configuring Salesforce to Trust CloudAccess," on page 68
- Section 6.3, "Configuring the Connector for Salesforce," on page 68
- Section 6.4, "Configuring Appmarks for Salesforce," on page 70
- Section 6.5, "Configuring Multiple Connectors for Salesforce," on page 71
- Section 6.6, "Using SSO to Salesforce on Mobile Devices," on page 71
- Section 6.7, "Configuring Delegated Authentication in Salesforce," on page 74
- Section 6.8, "Configuring the Salesforce Federation Identifier," on page 75

6.1 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 25.

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 6.2, "Configuring Salesforce to Trust CloudAccess," on page 68.

❒ 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.

6.2 Configuring Salesforce to Trust CloudAccess

NetIQ recommends 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`.

6.3 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 6.7, "Configuring Delegated Authentication in Salesforce," on page 74.

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 6.2, "Configuring Salesforce to Trust CloudAccess," on page 68.

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 6.7, "Configuring Delegated Authentication in Salesforce," on page 74.

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

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

4 Drag and drop the connector for Salesforce from the **Applications** palette to the **Applications** panel.

The Configuration window opens automatically for the initial configuration. To view or reconfigure the settings later, click the connector icon, then click **Configure**.

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 25.

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 6.8, "Configuring the Salesforce Federation Identifier," on page 75.

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

For more information, see Section 6.4, "Configuring Appmarks for Salesforce," on page 70.

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

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

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

21 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 *NetIQ® CloudAccess and MobileAccess Installation and Configuration Guide*.

22 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 6.6, "Using SSO to Salesforce on Mobile Devices," on page 71.

6.4 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 Section 2.5, "Configuring Appmarks for Connectors," on page 29.

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. In order to free a license for a paid account, you must de-activate the user.

6.5 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
```

6.6 Using SSO to Salesforce on Mobile Devices

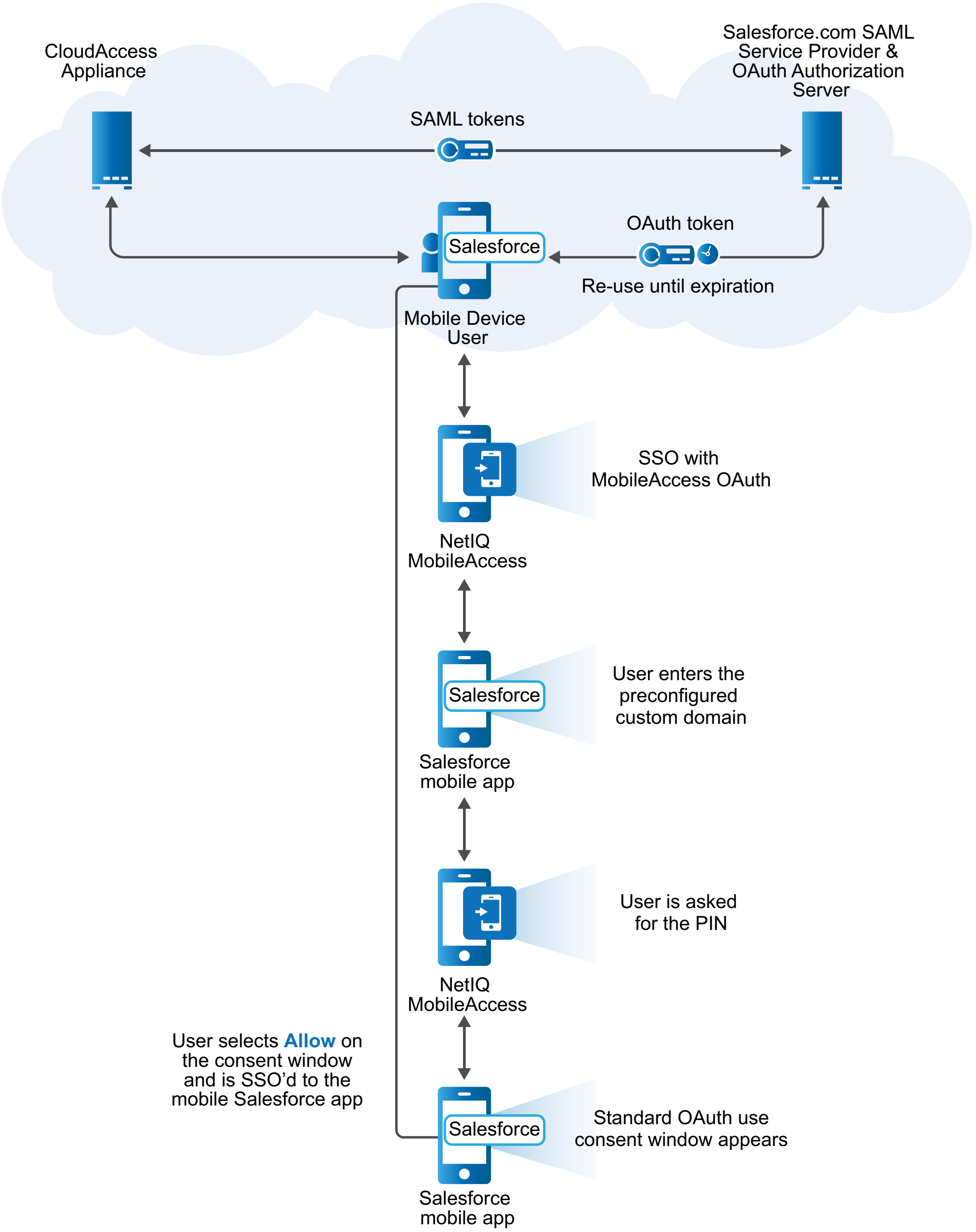
Salesforce now offers a combination of OAuth and SAML functionality to provide seamless SSO facilities not only for web browsers, but also desktop and mobile applications. By using OAuth to enable users to connect applications to their accounts, and leveraging SAML for the authentication of that connection, 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.

6.6.1 Understanding the Mobile SSO Process

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

Figure 6-1 *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 NetIQ 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 NetIQ 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 NetIQ MobileAccess login screen appears in step 4 and the user must enter credentials.

6.6.2 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 6.3, "Configuring the Connector for Salesforce," on page 68.
- ❒ 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 Section 2.5, "Configuring Appmarks for Connectors," on page 29.
- ❒ SAML SSO works on mobile devices only if the NetIQ MobileAccess app is also installed and configured on the device. Without MobileAccess installed on mobile devices, only delegated authentication using SAML is available.

  For more information, see "Installing MobileAccess on a Mobile Device" in the *NetIQ® CloudAccess and MobileAccess Installation and Configuration Guide*.

6.7 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 NetIQ 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 6.3, "Configuring the Connector for Salesforce," on page 68.

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 Before Provisioning** option, see Section 2.4, "How CloudAccess Provisions User Accounts," on page 25.

3 Configure a connector for Salesforce in CloudAccess as described in section Section 6.3, "Configuring the Connector for Salesforce," on page 68, 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`.

6.8 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 6.3, "Configuring the Connector for Salesforce," on page 68.

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 *NetIQ® CloudAccess and MobileAccess 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.

7 Connector for NetIQ Access Manager (SAML 2.0)

The connector for NetIQ Access Manager provides federated single sign-on access to Access Manager with SAML 2.0 through CloudAccess. It does not support provisioning. The connector allows CloudAccess to authenticate a user against your identity sources and to share this authentication with Access Manager in order to establish the user's session. You can also configure CloudAccess to provide MobileAccess features for Access Manager users.

CloudAccess includes this connector with the appliance. The connector appears automatically on the Applications palette of the Admin page. After you configure the connector, you must also configure Access Manager to work with the connector. Although you typically install the Access Manager Identity Server as an identity provider (IdP), you must configure it to be a service provider (SP) that consumes authentication information from CloudAccess.

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

- Section 7.1, "Requirements for the Connector for Access Manager," on page 77
- Section 7.2, "Configuring the Connector for Access Manager," on page 78
- Section 7.3, "Configuring Access Manager to Use CloudAccess as an Identity Provider," on page 80
- Section 7.4, "Configuring Appmarks for Protected Resources in Access Manager," on page 82

7.1 Requirements for the Connector for Access Manager

❒ A CloudAccess appliance, installed and configured. MobileAccess configuration is optional, depending on your user authentication needs.

❒ A NetIQ Access Manager 4.0.*x* 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" in the *NetIQ Access Manager Setup Guide*. 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://<access_manager_identity_server_dns_name>/nidp/saml2/metadata
```

❒ The SSL signing certificate from Access Manager.

IMPORTANT: The configuration 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" in the *NetIQ Access Manager Administration Console Guide*.

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).

7.2 Configuring the Connector for 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.

Before you begin, ensure that your system meets the requirements in Section 7.1, "Requirements for the Connector for Access Manager," on page 77.

To configure the connector for Access Manager:

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

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

3. Drag and drop the SAML 2.0 connector for NetIQ Access Manager from the **Applications** palette to the **Applications** panel.

   The Configuration window opens automatically for the initial configuration. To view or reconfigure the settings later, click the connector icon, 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 identity source 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 the **Appmarks** tab, then review and edit the default settings for the appmark.

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 As the Access Manager administrator, configure the SAML 2.0 federation for CloudAccess.

For more information, see Section 7.3, "Configuring Access Manager to Use CloudAccess as an Identity Provider," on page 80.

Use the information from the **Federation Instructions** in Step 6 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.

12 In the CloudAccess administration console, 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 *NetIQ® CloudAccess and MobileAccess Installation and Configuration Guide*.

7.3 Configuring Access Manager to Use CloudAccess as an Identity Provider

After configuring the connector, you must configure Access Manager to use CloudAccess or MobileAccess as a trusted external identity provider.

IMPORTANT: The following checklist identifies the tasks to perform in NetIQ Access Manager. For step-by-step instructions with sample values, see *Using NetIQ® CloudAccess as a Trusted Identity Provider for NetIQ® Access Manager*.

❒ **Create 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 identity source, 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.

For more information, see "Configuring Attribute Sets" in the *NetIQ Access Manager Identity Server Guide*.

❒ **Create an attribute matching expression to use for the identity provider user identification.**

When Access Manager receives an assertion from CloudAccess, it uses the attributes in the assertion to match the user to an identity in your identity sources. You must know which attributes in the identity source are used to uniquely identify the users. Use the same attribute that you specified in the attribute set, such as `Ldap Attribute:workforceID`.

For more information, see "Configuring User Matching Expressions" in the *NetIQ Access Manager Identity Server Guide*.

❒ **Create an external identity provider for SAML 2.0 that represents CloudAccess.**

CloudAccess acts as an external identity provider to Access Manager. You must establish a trust between them so that two user accounts can be associated with each other without the sites exchanging user data.

For more information, see "Creating a Trusted Service Provider for SAML 2.0" in the *NetIQ Access Manager Identity Server Guide*.

❒ **Create an external authentication contract for the identity provider that represents CloudAccess.**

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. 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.

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, see "Configuring Authentication Contracts" in the *NetIQ Access Manager Identity Server Guide*.

❒ **Configure a SAML 2.0 authentication request for the identity provider that represents CloudAccess.**

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, see "Configuring an Authentication Request for an Identity Provider" in the *NetIQ Access Manager Identity Server Guide*.

❒ **(Conditional) If you use an eDirectory identity source and the protected applications require a password, configure 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.

For more information, see "Configuring Password Retrieval" in the *NetIQ Access Manager Identity Server Guide*.

❐ **Configure a user identification method to use for the identity provider that represents CloudAccess.**

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, see "Selecting a User Identification Method for Liberty or SAML 2.0" in the *NetIQ Access Manager Identity Server Guide*.

❐ **Configure attributes for the identity provider that represents CloudAccess.**

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.

For more information, see "Configuring the Attributes Obtained at Authentication" in the *NetIQ Access Manager Identity Server Guide*.

❐ **Assign the external authentication contract to the 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" in the *NetIQ Access Manager Access Gateway Guide*.

7.4 Configuring Appmarks for Protected Resources in Access Manager

After you have configured the connector for Access Manager and single sign-on SAML 2.0 federation between Access Manager and CloudAccess, you can configure appmarks for protected resources in Access Manager.

The default appmark for the connector for Access Manager uses the Destination URL field from the configuration. If you did not specify the Destination URL, you will end up at the Access Manager home page for the default appmark.

For more information, see Section 2.5, "Configuring Appmarks for Connectors," on page 29.

8 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

8.1 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 and drop the **Bookmark** connector from the **Applications** palette to the **Applications** panel.

The Configuration window opens automatically for the initial configuration. To view or reconfigure the settings later, click the connector icon, then click **Configure**.

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 using the standard “pull-to-refresh” action on the Applications page in the MobileAccess app. (This action is used in mail and other common applications on the mobile device.) How users access the bookmark appmark depends on how you configured the **Launch with** option.

9 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 9.1, "Configuring the OAuth2 Client Application," on page 85
- Section 9.2, "Configuring the Connector for OAuth2 Resources," on page 86
- Section 9.3, "Supported OpenID Connect Schema," on page 87

9.1 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.

9.2 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 and drop the **OAuth Resources** connector from the **Applications** palette to the **Applications** panel.

   The Configuration window opens automatically for the initial configuration. To view or reconfigure the settings later, click the connector icon, then click **Configure**.

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 *NetIQ® CloudAccess and MobileAccess 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.

9.3 Supported OpenID Connect Schema

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

Table 9-1 *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.

10 Connector for Simple Proxy

The connector for Simple Proxy provides reverse proxy access to your enterprise 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. You can configure the connector to protect access to the document root of the web server, or to protect access only to a path within the document root of the web server.

Every web server is different. Two common simple proxy scenarios are:

- **Simple website:** If your web server provides a single web service, you can protect access to the entire site by creating a connector for Simple Proxy. The connector points to the document root of the web server.
- **Multiple-service website:** If your server provides multiple web services, you can create a separate connector for Simple Proxy for each destination web service. Each connector points to a different independent path within the document root of the web server.

If the web service requires user identity information to control access or content, you can configure the connector to inject the authenticated user’s identity attributes in query strings and HTTP headers sent to the web service. Because the connector cannot send passwords, the connector cannot be used to provide single sign-on for web services that require passwords for access. It does not support provisioning.

CloudAccess includes this connector with the appliance. The connector is included automatically on the Applications palette of the Admin page. Each cluster supports multiple connectors for Simple Proxy.

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

- Section 10.1, “Requirements for Simple Proxy,” on page 89
- Section 10.2, “Viewing or Customizing the Attributes for Identity Injection,” on page 90
- Section 10.3, “Configuring the Connector for Simple Proxy,” on page 93

10.1 Requirements for Simple Proxy

The connector for Simple Proxy enables reverse proxy access to an enterprise web server behind your firewall. It can support web services that employ user identity information to control access or display if you enable the identity injection policies that insert an authenticated user’s identity attributes in query strings or headers of requests it sends to the web server. For more information, see Section 10.2, “Viewing or Customizing the Attributes for Identity Injection,” on page 90.

For each proxy web service, the web service’s content should be self-contained in that path. If the service depends on files that reside in parallel paths on the web server, you can specify a path at a higher level in the document root’s directory structure, or reorganize the site’s contents as needed.

The connector for Simple Proxy does not support the following:

- **Protected resources that require a password:** This proxy solution cannot be used with protected web services or applications that require an LDAP password to be included in the identity injection. The appliance cannot send a user's password for a proxy application to the back end web service.

  If the web server needs the user's password, you must find a workaround. For example, you could specify a static string that is accepted for all users.
- **Site redirects:** This proxy solution does not support site redirects to locations outside the protected path. It cannot follow paths to alternate websites.

IMPORTANT: The Access Gateway for NetIQ Access Manager provides solutions for more complex reverse proxies that support password injection and redirects. For more information, see "Managing Reverse Proxies and Authentication" in the *NetIQ Access Manager Access Gateway Guide*.

Before you configure a connector for Simple Proxy, ensure that your setup meets the following requirements:

❐ A CloudAccess system, installed and configured.

❐ A web server, configured and running behind the corporate firewall. Ensure that you have configured the authentication procedures and identity injection policy for the web service.

You need the following information:

- The primary DNS name or IP address of the web server.
- Alternative DNS names or IP addresses for the web server, if any.
- The port number that the web server uses to listen for requests, such as 8080 (non-secure) or 8443 (secure SSL).
- If the web server requires it, secure communications with HTTPS.

  If you use HTTPS, the value that you specify for the web server's DNS name or IP address in the connector must match the CN in the web server's SSL certificate.

❐ Determine which web services you need to protect for your web server, and which users require access to each one.

10.2 Viewing or Customizing the Attributes for Identity Injection

The connector for Simple Proxy can inject an authenticated user's identity attributes in query strings and headers of communications sent from the appliance to the destination web service. The web server might use this information to determine whether the user should have access to the resource. It can also use the identity information to customize content on the web page. For example, when a user whose first name is Joe (as specified in the identity source) navigates to the destination web page, he might see "Welcome: Joe" at the top of his browser window.

- Section 10.2.1, "Understanding Identity Attributes," on page 91
- Section 10.2.2, "Viewing Identity Attribute Mappings to Identity Source Attributes," on page 92
- Section 10.2.3, "Configuring Custom Identity Attributes," on page 93

10.2.1 Understanding Identity Attributes

In the connector for Simple Proxy, you can enable or disable the following identity injection policies. Both policies are enabled by default.

- **Inject Identity in Query:** If you enable this option, when a user navigates to the connector's destination web service, the service receives all of the user's identity attributes in the query string.

  WARNING: Injecting attributes in the query string could exceed the maximum URL length of 2083 characters.

- **Inject Identity in Header:** If you enable this option, when a user navigates to the connector's destination web service, the service receives all of the user's identity attributes as custom headers.

If you enable an injection policy, the connector sends all of the user's identity attributes, even if the values are unavailable (empty). For some applications, this is still useful information and the web service can use it to make access or display decisions.

WARNING: If you use HTTP for communications between the connector and the web service, the injected identity attributes are available as clear text to network packet sniffers.

Although the proxy service runs behind the firewall, consider configuring the connector's web service URL with HTTPS to protect the communication stream to the web service. If you use HTTPS, the value that you specify for the web server's DNS name or IP address in the connector must match the CN in the web server's SSL certificate.

The attribute values in the query strings parameters or header parameters sent to the web server are based on the following options in the identity source user interface:

Identity Source Parameter	Query String Parameter	Header Parameter
ID	ID	X-ID
Email	Email	X-Email
User name	UserName	X-UserName
First name	FirstName	X-FirstName
Middle name	MiddleName	X-MiddleName
Last name	LastName	X-LastName
Full name	FullName	X-FullName
Preferred name	PreferredName	X-PreferredName
Generational qualifier	GenerationalQualifier	X-GenerationalQualifier
Gender	Gender	X-Gender
Phone	Phone	X-Phone
Birthdate	BirthDate	X-BirthDate
Street address	StreetAddress	X-StreetAddress
City	City	X-City
State	State	X-State
ZIP code	ZipCode	X-ZipCode
Country	Country	X-Country
Language	Language	X-Language
Identity Type	IdentityType	X-IdentityType
X-Custom1	XCustom1	X-XCustom1
X-Custom2	XCustom2	X-XCustom2
X-Custom3	XCustom3	X-XCustom3
X-Custom4	XCustom4	X-XCustom4
X-Custom5	XCustom5	X-XCustom5

The IdentityType parameter for query strings and headers indicates the type of identity source that the appliance uses to authenticate the user, such as Active Directory, eDirectory, Self-Service User Store (SSUS), and JDBC.

10.2.2 Viewing Identity Attribute Mappings to Identity Source Attributes

In your identity source, the identity attributes are mapped to identity source attributes. You can view mappings in your identity source connector. For example, in eDirectory, `ID` is mapped to the `guid` attribute, `User name` is mapped to the `cn`, and so on. You can change these mappings as needed for your environment, but any changes you make are global. You cannot change them on a per proxy or app basis.

To view identity attribute mappings in an identity source:

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

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

2. In the **Identity Sources** panel, click the identity source, then click **Configure**.
3. Expand **Advanced Options**.
4. In the **Attribute Mappings** section, expand **Default** to view the list of the mappings of identity attributes to identity source attributes.

5. If you modify the settings, click **OK** to save your changes, and then click **Apply** on the Admin page.

   Do not continue until the changes are applied to all nodes of the appliance cluster.

6. Repeat this process for each identity source that manages users who will access the destination web server.

10.2.3 Configuring Custom Identity Attributes

An identity injection sends all identity attributes. You cannot specify only a subset of attributes, add attributes, or remove attributes. However, you can map the `X-Custom`<1-5> attributes to attributes in your identity source. Ensure that you map the appropriate identity source attribute to each custom attribute across all of the identity sources for users who will access the destination web server.

To configure custom identity attributes in your identity source:

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

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

2. In the **Identity Sources** panel, click the identity source, then click **Configure**.
3. Expand **Advanced Options**, then use the **Attribute Mappings** section to map custom attributes (`X-Custom`<1-5>) to attributes in your identity source.
4. Click **OK** to save your changes, and then click **Apply** on the Admin page.

   Do not continue until the changes are applied to all nodes of the appliance cluster.

5. Repeat this setup for each identity source that manages users who will access the destination web server.

10.3 Configuring the Connector for Simple Proxy

Each connector for Simple Proxy can protect only a single web location. If the connector is set to protect the document root, then users can access all files served by the website. If the connector is set to protect a path under the document root, users can access only those files that reside in the path or its subdirectories.

The connector protects access to the web service based on its Appmarks settings. You can allow public access so that all users have access to the web service. You can alternatively deny public access and grant access to users of specific identity source roles by mapping authorization policies for them. Authorization policies dynamically control visibility of appmarks on the landing page for users after authentication. They enforce access when a user attempts to access a protected resource with an appmark, or when they directly browse the website.

To configure the connector for Simple Proxy:

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

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

2. Drag and drop the connector for Simple Proxy from the **Applications** palette to the **Applications** panel.

   The Configuration window opens automatically for the initial configuration. To view or reconfigure the settings later, click the connector icon, then click **Configure**.

3 In the Configuration window on the **Configuration** tab, provide the following information:

Display name: Specify the display name for the reverse proxy service. This name is also the default name of the appmark that appears in the user interface.

Local path: Specify a unique path on the appliance that will be used in the URL to associate traffic for the remote web service, such as `/myservice` or `servicexyz`. The path will be appended to the DNS name of the cluster for accessing the resource, and will be removed from the request before forwarding it to the web server.

The local path must be unique across all connectors for Simple Proxy that you configure on the appliance. You can use alphanumeric characters a to z and 0 to 9, forward slashes (/), hyphens (-), and underscores (_). Spaces, uppercase characters, and other special characters are not supported. The path is not case sensitive. There is no length limit, but you should consider length restrictions for URLs and file system pathnames when you specify the character string.

Connects to: Specify the URL of the destination web service that you want to protect.

You can use HTTP (not secure) or HTTPS (secure) in the URL, depending on requirements of the web server. If you use HTTPS, the value that you specify for the DNS name or IP address must match the CN in the web server's SSL certificate. The connector automatically finds the SSL certificate and installs it for you if the URL uses HTTPS.

You can specify the IP address or DNS name of the web server. Specify the port number if it is needed to access the location.

Do one of the following:

- Specify the root of the web server in order to protect all resources in the document root of the web server, including its subdirectories and their content.

  For example, you can specify the URL in any of the following formats:

  ```
  http://10.20.30.40
  http://10.20.30.40:8080
  https://myweb.example.com
  https://myweb.example.com:8443
  ```

- Specify a path within the document root of the web server in order to protect only the resources in that path, including its subdirectories and their content.

  For example, you can specify the URL in any of the following formats:

  ```
  http://10.20.30.40/path_to_protect
  http://10.20.30.40:8080/path_to_protect
  https://myweb.example.com/path_to_protect
  https://myweb.example.com:8443/path_to_protect
  ```

Inject Identity in Query: Select this option to include the user identity attributes in the query strings that are sent to the **Connects to** URL. For more information, see Section 10.2, “Viewing or Customizing the Attributes for Identity Injection,” on page 90.

WARNING: Injecting attributes in the query string could exceed the maximum URL length of 2083 characters.

Inject Identity in Headers: Select this option to include the user identity attributes in the headers that are sent to the **Connects to** URL. For more information, see Section 10.2, “Viewing or Customizing the Attributes for Identity Injection,” on page 90.

4 Expand **Advanced Options**, then configure the **Rewriter Options**:

The rewriter parses and searches the web content that passes through the appliance for URL references that qualify to be rewritten. URL references are rewritten when they meet the following conditions:

Strip Local path from query string: Enables URL references specified in the query strings to be rewritten with the published DNS name.

Strip Local path from POST data: Enables URL references specified in the post data to be rewritten with the published DNS name.

Strip Local path from REFERRER header: Enables URL references specified in the referrer headers to be rewritten with the published DNS name.

Alternative Host Names: URL references that match entries in this list are rewritten with the published DNS name. You can use any of the following formats. The entries are not case sensitive.

```
site.example.com
myhostname
10.10.2.10
http://<dns_name_or_ip_address>
http://<dns_name_or_ip_address>:port
https://<dns_name_or_ip_address>
https://<dns_name_or_ip_address>:port
```

You need to include names in this list if your web servers have the following configurations:

- If you have a cluster of web servers that are not sharing the same DNS name, you need to add their DNS names to this list.
- If your web server obtains content from another web server, the DNS name for this additional web server needs to be added to the list.
- If the web server listens on one port (for example, 80), and redirects the request to a secure port (for example, 443), the DNS name needs to be added to the list. This allows the response to be sent in the format that the user expects.
- If an application is written to use a private hostname, you need to add the private hostname to the list. For example, `http://<hostname>/index.html`.

5. Click the **Appmarks** tab, then review and edit the default settings for the appmark.
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, 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 *NetIQ® CloudAccess and MobileAccess Installation and Configuration Guide*.

11 Connectors for Basic SSO

Each connector for Basic Single Sign-On (SSO) provides forms-based single sign-on to a destination web service or application through CloudAccess. It meets the specific interactive and content requirements for logging in to the website. A connector works with the Basic SSO extension for the Chrome browser to securely collect, store, retrieve, and replay the user's authentication information for that website. For information about how CloudAccess keeps the user's credentials secure, see *Section 11.3, "How CloudAccess Stores Credentials Securely with Basic SSO," on page 99*.

CloudAccess provides many connectors for Basic SSO that you can download from NetIQ Downloads (https://dl.netiq.com/). NetIQ CloudAccess adds new connectors for download as they become available.

IMPORTANT: Please contact NetIQ 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 NetIQ Access Connector Toolkit. For more information, see Chapter 3, "Creating Custom Connectors," on page 37.

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:

11.1 Requirements for Using Basic SSO with Websites

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

- The destination website's login page uses HTML Forms as the main point of interaction with the user.
- The destination website requires the user's password to be sent for logging in.
- The destination website 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 destination websites only through a Chrome web browser running on a desktop or laptop computer. They do not support access from mobile devices.

❒ The NetIQ Basic SSO extension is compatible only with the Chrome web browser. A user must install the extension in the Chrome browser one time on each desktop or laptop they use to access the Basic SSO websites.

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

11.2 Understanding the Basic SSO Service

Each connector for Basic SSO works with the NetIQ Basic SSO extension for the Chrome browser, running on the user's computer. It collects, saves, retrieves, and replays a user's login credentials and metadata in a format that the site requires on its login page. The user must install the Basic SSO extension to take advantage of the single sign-on capability.

On successful site login, CloudAccess allows the user to specify whether to save the credentials for the website. If the user approves, CloudAccess securely stores the user's credentials for the website in the internal credential store. It does not save them locally on the user's computer. The user's credentials are available for single sign-on to the application through CloudAccess from any computer where the Basic SSO extension for the Chrome browser has been installed.

In subsequent CloudAccess sessions, the user can log in with enterprise credentials to CloudAccess and access the destination website without providing the additional credentials. CloudAccess securely retrieves and replays the user's site login information for an automatic login on behalf of the user. Thus, the user has the experience of single sign-on.

Typically, users have a different login user name and password for their individual accounts on destination websites. CloudAccess can store only one account for each destination website for the user.

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 destination website to modify the user name and password as needed.

If the user changes the user name or password to the account, or if the user cancels the account, the user's stored credentials are no longer valid. The automatic login fails, and CloudAccess removes the user's old credentials for the website. CloudAccess redirects the user to the website's login page where the user can log in with new credentials and save them if desired.

If the user wants to remove credentials for a website from the credential store, the user can use the website's interface to change the password and exit the site. When the user accesses the application again, the login fails, and CloudAccess removes the user's old credentials for the website from the credential store. CloudAccess redirects the user to the website's login page where the user can log in with new credentials, and then choose to not save credentials for that website.

If a user uninstalls the Basic SSO extension, it does not affect the user's credentials stored in the credential store. However, the user cannot take advantage of the single sign-on capability. If the user installs the extension again, the single sign-on capability starts working again.

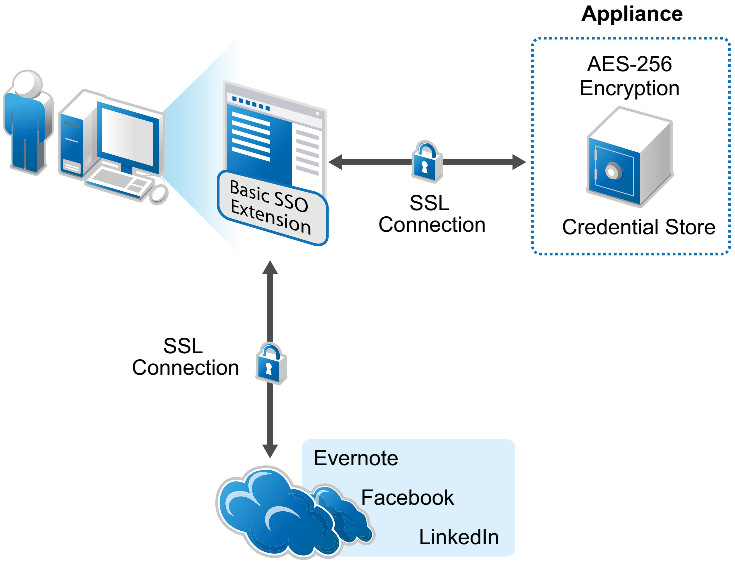
NOTE: The MobileAccess app does not currently support a single-sign experience for links to websites configured with connectors for Basic SSO. The link takes the user to the website, but the user must enter the additional credentials.

11.3 How CloudAccess Stores Credentials Securely with Basic SSO

Basic SSO describes a level of authentication that CloudAccess supports. Basic SSO records and stores users' login credentials to be re-used when users authenticate to the destination site. Users must enter their credentials once, and then CloudAccess is able to capture and store them for use again. Only the user who stores the credentials can access the various user names and passwords for Basic SSO websites that are stored for that user's account in the credential store. No other users can access the information in its unencrypted format, including administrators.

CloudAccess protects user credentials through an SSL connection and AES-256 encryption on the appliance. Figure 11-1 depicts how CloudAccess stores the credentials securely.

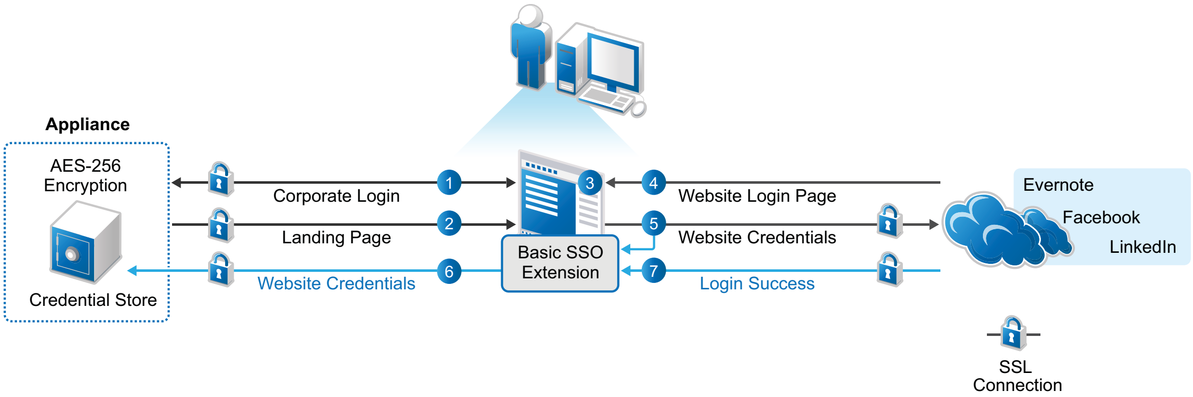
Figure 11-1 *Basic SSO Security*



Basic SSO connectors work with the Basic SSO extension for the Chrome browser running on the user's computer to securely collect, store, retrieve, and replay the user credentials for a destination website. Users must log in to the website once in order for the extension to capture and store the credentials in the CloudAccess credential store. The user can choose whether to store the credentials for each destination website. If the user does not allow credentials to be saved for a website, the user must enter the site's credentials for each session.

Figure 11-2 depicts the user experience when the user clicks the appmark for a Basic SSO application.

Figure 11-2 User's First-Time Login to the Website with Basic SSO



The following describes the experience for Basic SSO the first time the user accesses the app:

1. In a Chrome browser, the user logs in to the CloudAccess login page using their corporate credentials.
2. The user sees the available applications on the landing page.
3. The user clicks the appropriate application icon.

   If the Basic SSO extension for the Chrome browser is not installed on the computer:

   a. The connector prompts the user to install the Basic SSO extension.
   b. The user accepts the prompt, and the appliance opens the Google Play Store in a new tab.
   c. The user installs the Basic SSO extension, then closes the Google Play Store tab to continue.
   d. The user returns to the landing page and clicks the appropriate application icon again.
4. A new tab opens for the login page of the application.
5. The user enters their user name and password for the destination website.

   The user must enter this separate user name and password once.
6. The extension asks if the user wants the credentials to be saved by CloudAccess, and the user allows the credentials to be saved.

   a. The extension captures the user name and password, and sends them to CloudAccess over an SSL connection.

      The extension obfuscates the user name and password with Base64 encoding before transmission.
   b. CloudAccess encrypts the site-specific credentials with AES-256 encryption, and then stores the encrypted data in the credential store that is part of the appliance.

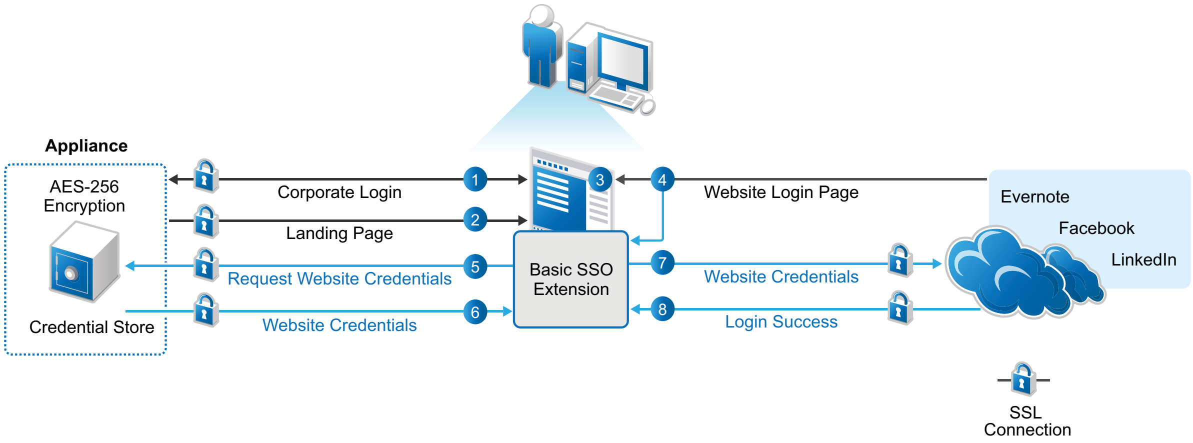
      The appliance encrypts the user name and password with an encryption key that is unique per user.
7. The website returns a success or failure indicator for the login.

   If the login succeeds, the browser opens to the application's website over an SSL connection.

   If the login fails, the browser returns the user to the website's login page to try again, and the extension requests that CloudAccess remove the saved credentials.

After the user allows the password to be stored securely, the user experiences single-sign-on access to the application in subsequent sessions. Figure 11-3 depicts the user experience when the user clicks the appmark for a Basic SSO application and the user's credentials are available in the credentials store.

Figure 11-3 *User's Single Sign-On Access to a Website with Basic SSO*



The following describes the experience for Basic SSO after the user stores credentials:

1. The user logs in to the CloudAccess login page using their corporate credentials.
2. The user sees the available applications on the landing page.
3. The user clicks the appropriate application icon.
4. A new tab opens for the login page of the application.
5. The Basic SSO extension requests that CloudAccess retrieve the user's user name and password for the site from the credential store.
6. CloudAccess retrieves the site-specific encrypted credentials from the credential store, decrypts them, and then sends the user name and password to the application's website over an SSL connection.

   CloudAccess obfuscates the user name and password with Base64 encoding before transmission.
7. CloudAccess logs in the user to the application's website. To the user, it appears as a single sign-on experience.

   If the user changes their login credentials for the destination website, the user will be prompted to log in again and the new credentials will be stored using the same process as for the initial setup.
8. The website returns a success indicator for the login, and the browser opens to the application's website over an SSL connection.

11.4 Configuring a Connector for Basic SSO

You can import and configure as many of these connectors as you need on your CloudAccess system.

To configure one or more connectors for Basic SSO:

1. Download the connector for Basic SSO for a destination website, or create a custom Basic SSO connector for a desired website.
2. Log in as an administrator to the CloudAccess administration console:

   ```
   https://appliance_dns_name/appliance/index.html
   ```
3. In the toolbar, click the **Tools** icon, click **Import Connector Template**, then browse to and select the connector template you want to import.
4. Drag and drop the imported connector from the **Applications** palette to the **Applications** panel.

   The Configuration window opens automatically for the initial configuration. To view or reconfigure the settings later, click the connector icon, then click **Configure**.
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 3 to Step 6 to configure additional connectors for Basic SSO at this time.
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 *NetIQ® CloudAccess and MobileAccess Installation and Configuration Guide*.

12 Connector for Accellion (SAML 2.0)

The connector for Accellion provides federated single sign-on (SSO) access to Accellion with SAML 2.0 through CloudAccess. It does not support provisioning. The connector allows CloudAccess to authenticate a user against your identity sources and to share this authentication with Accellion in order to establish the user's session.

You can download the connector for Accellion from NetIQ Downloads (https://dl.netiq.com/). You must import the connector to CloudAccess, configure it to work with your Accellion account, and then map policies to set entitlements to Accellion for your users. You must also configure Accellion to work with the connector.

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

- Section 12.1, "Requirements," on page 103
- Section 12.2, "Configuring the Connector," on page 104

12.1 Requirements

Verify that you meet the following requirements before you import the connector:

❒ 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).

❒ An enterprise Accellion account.

❒ Administrator access to an enterprise Accellion account. An understanding of Accellion and its account management tools are presumed.

❒ An Accellion user account for each user who wants to authenticate to Accellion through the CloudAccess single sign-on service. The connector for Accellion does not provision user accounts.

❒ The location in the Accellion 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 Accellion for CloudAccess. This information includes the metadata; a signing certificate for the appliance; the field values to use; and other guidance.

❒ The Accellion domain name for your enterprise account and the application ID. The Accellion administrative URL contains both elements.

```
http://domain_name.accellion.net/courier/application_id/index.html
```

❒ (Optional) An X.509 signing certificate from Accellion is required to support single logout. Communications use SSL regardless of whether you provide this certificate.

12.2 Configuring the Connector

After you import the connector, you must configure it to work with your enterprise Accellion account.

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

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

2 Drag and drop the connector for Accellion from the **Applications** palette to the **Applications** panel.

The Configuration window opens automatically for the initial configuration. To view or reconfigure the settings later, click the connector icon, then click **Configure**.

3 On the **Configuration** page, specify the configuration properties.

Use the domain name and account ID from the Accellion administration URL. The signing certificate from Accellion is optional.

4 Under **Assertion Attribute Mappings**, map the SAML Assertion attributes to the appropriate attributes in your identity source.

5 Expand the **Federation Instructions**, then copy and paste the instructions into a text file to use during the Accellion 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.

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

For more information, see Section 2.5, “Configuring Appmarks for Connectors,” on page 29.

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 cluster.

10 Log in to Accellion as the enterprise account administrator, then configure the SAML 2.0 federation for CloudAccess in the Accellion 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** in the toolbar, then perform policy mapping to specify entitlements for identity source roles (groups).

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

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

```
https://appliance_dns_name
```

13 Connector for ADFS (SAML 2.0)

The connector for Active Directory Federation Services (ADFS) provides federated single sign-on (SSO) access to ADFS with SAML 2.0 through CloudAccess. It does not support provisioning. The connector allows CloudAccess to authenticate a user against your identity sources and to share this authentication with ADFS in order to establish the user's session.

You can download the SAML 2.0 connector for ADFS from NetIQ Downloads (https://dl.netiq.com/). You must import the connector to CloudAccess, configure it to work with your ADFS system, and then map policies to set entitlements to ADFS for your users. You must also configure ADFS to work with the connector.

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

- Section 13.1, "Requirements," on page 105
- Section 13.2, "Configuring the Connector," on page 106
- Section 13.3, "Troubleshooting Certificate Errors," on page 107
- Section 13.4, "Connecting to SharePoint," on page 107

13.1 Requirements

Verify that you meet the following requirements before you import the connector:

❒ 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).

❒ An ADFS 2.0 system, installed and configured.

❒ Administrator access to the ADFS system. An understanding of ADFS and its management tools are presumed.

❒ An ADFS user account for each user who wants to authenticate to ADFS through the CloudAccess single sign-on service. The connector for ADFS does not provision user accounts.

❒ The location in the ADFS 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 ADFS for CloudAccess. This information includes the metadata; a signing certificate for the appliance; the field values to use; and other guidance.

❒ The metadata file from the ADFS 2.0 system.

```
https://adfsserver/FederationMetadata/2007-06/FederationMetadata.xml
```

You will need the following information from the metadata file:

- **Assertion Consumer Service URL:** The value in the **AssertionConsumerService** field with the HTTP-POST binding.

- **EntityID:** The value in the **entityID** field.
- **Logout URL:** The value in **SingleLogoutService Location** field with the HTTP-POST binding.

❒ (Optional) An X.509 signing certificate from ADFS is required to support single logout. Communications use SSL regardless of whether you provide this certificate.

13.2 Configuring the Connector

After you import the connector, you must configure it to work with your ADFS system.

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

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

2 Drag and drop the SAML 2.0 connector for ADFS from the **Applications** palette to the **Applications** panel.

The Configuration window opens automatically for the initial configuration. To view or reconfigure the settings later, click the connector icon, then click **Configure**.

3 On the **Configuration** page, specify the configuration properties.

Use the information from the ADFS metadata file. The signing certificate from ADFS is optional.

4 Under **Assertion Attribute Mappings**, map the SAML Assertion attributes to the appropriate attributes in your identity source.

5 Expand the **Federation Instructions**, then copy and paste the instructions into a text file to use during the ADFS 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.

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

For more information, see Section 2.5, “Configuring Appmarks for Connectors,” on page 29.

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 cluster.

10 Log in to ADFS as the ADFS administrator, then configure the SAML 2.0 federation for CloudAccess in the ADFS 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** in the toolbar, then perform policy mapping to specify entitlements for identity source roles (groups).

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

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

```
https://appliance_dns_name
```

13 (Optional) To allow Service Provider-initiated login, you must specify the Name ID format on the ADFS side. To do this, run the following PowerShell command:

```
Set-ADFSClaimsProviderTrust -TargetName Display Name from Claims
Provider Trust -RequiredNameIdFormat
urn:oasis:names:tc:SAML:1.1:nameid-format:emailAddress
```

14 (Optional) If you want users to connect to SharePoint, proceed to Section 13.4, "Connecting to SharePoint," on page 107.

13.3 Troubleshooting Certificate Errors

If you use a self-signed certificate and certificate chain errors occur, run the following PowerShell command:

```
Set-ADFSClaimsProviderTrust -TargetName Display Name from above -
SigningCertificateRevocationCheck None
```

13.4 Connecting to SharePoint

With additional configuration, the SAML 2.0 connector for ADFS allows users to single sign-on through CloudAccess to SharePoint as well as ADFS.

This section describes how you can leverage the claims-based single sign-on capabilities of ADFS and SharePoint to set up a hub model of federation through ADFS. In this hub, CloudAccess has a trusted relationship with ADFS as the identity provider, and ADFS has a trusted relationship with SharePoint as a claims-based federation provider. SharePoint accepts the claims-based assertions, and allows users to access federated SharePoint web applications. Using roles for claim-based single sign-on makes it easier for SharePoint site administrators to map role and organization claims to SharePoint groups.

To set up the relationships, you define the roles in the connector for ADFS that ADFS and SharePoint will use for the claims-based single sign-on. The connector adds the role information to the identity information in assertions that it sends to ADFS.

In ADFS, you configure claims rules that look for the email address and role of users, and then transform them for use by SharePoint. ADFS applies rules to the assertions from CloudAccess to transform them into role claims that the SharePoint web applications understand, and sends the role claims to SharePoint.

In SharePoint, you configure its Person Picker to look for the roles in the assertions from ADFS. SharePoint validates the assertion information, stores the information in its token cache, and issues a session cookie for the user. By default, SharePoint sets the session lifetime to be the same as the SAML token lifetime. In ADFS, you can specify the web single sign-on lifetime that determines the lifetime of the session cookie. Typically, the cookie expires when the user closes the browser window.

To set up this claims-based single sign-on federation hub:

- The CloudAccess administrator must modify the definition for the connector for ADFS to add two new roles to use for claims-based single sign-on, and then import and configure the modified connector.
- The ADFS administrator must configure a connection between SharePoint and ADFS, and define the rules for passing identity and role information from CloudAccess to SharePoint.
- The SharePoint administrator must modify the SharePoint People Picker to look for the roles in incoming assertions.
- The SharePoint administrator can add users to a SharePoint group based on the users' roles.

Use the following information to set up a federation relationship between SharePoint and CloudAccess that uses ADFS as a federation provider for SharePoint.

- Section 13.4.1, "Requirements," on page 108
- Section 13.4.2, "Adding Roles to the SAML 2.0 Connector for ADFS," on page 108
- Section 13.4.3, "Modifying Claims Rules in the ADFS System," on page 110
- Section 13.4.4, "Configuring the SharePoint People Picker to Use the Roles," on page 111
- Section 13.4.5, "Troubleshooting SharePoint Issues," on page 112

13.4.1 Requirements

Verify that you meet the following requirements:

- ❐ A CloudAccess appliance, installed and configured.
- ❐ One server with the following components installed:
  - ❐ Windows Server 2008 (or later) with the latest updates.
  - ❐ Active Directory with the latest updates.
  - ❐ ADFS 2.0 with the latest updates.
- ❐ A SharePoint 2010 (or later) server with the latest updates, installed in the same domain as the ADFS server.
  - ❐ The SharePoint server should be connected to the ADFS server.

    For information about connecting the servers, see the following references in the Microsoft TechNet Library:

    - *How to Configure ADFS v 2.0 in SharePoint Server 2010* (http://technet.microsoft.com/en-us/library/hh305235%28v=office.14%29.aspx).
    - *Configure SAML-based Claims Authentication with ADFS in SharePoint 2013* (http://technet.microsoft.com/en-us/library/hh305235%28v=office.15%29.aspx)
  - ❐ Roles enabled within the SharePoint system using PowerShell scripts.

13.4.2 Adding Roles to the SAML 2.0 Connector for ADFS

You must modify the definitions in a SAML 2.0 connector for ADFS template file to add roles that will be used when ADFS sends role claims to SharePoint. These instructions create two roles: an administrator role called `ADMIN` and a user role called `USER`.

- "Modifying the SAML 2.0 Connector for ADFS Template" on page 108
- "Importing the Modified Connector" on page 109
- "Configuring the Modified Connector" on page 110

Modifying the SAML 2.0 Connector for ADFS Template

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

1. Obtain a copy of the ZIP file for the SAML 2.0 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 "Importing the Modified Connector" on page 109.

Importing the Modified Connector

After you modify the SAML 2.0 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 SAML 2.0 connector for ADFS.

4 Click **Import**.

The Applications palette displays the modified SAML 2.0 connector for ADFS.

5 Proceed to "Configuring the Modified Connector" on page 110.

Configuring the Modified Connector

After you export and import the modified connector, you configure the connector by following the steps in Section 13.2, "Configuring the Connector," on page 106.

After you configure a SAML 2.0 connector for ADFS that supports SharePoint roles, you must modify ADFS and SharePoint to accept these roles. Proceed to "Modifying Claims Rules in the ADFS System" on page 110.

13.4.3 Modifying Claims Rules in the ADFS System

Before you begin, ensure that you have configured a connection between ADFS and SharePoint. In ADFS, you must define the claim rules for incoming assertions from CloudAccess and for outgoing assertions sent to SharePoint.

- "Adding Claims Rules for SharePoint Roles in Incoming Assertions" on page 110
- "Adding Claims Rules for Transforming Assertions for SharePoint" on page 111

Adding Claims Rules for SharePoint Roles in Incoming Assertions

You must add the ADFS claim rules between ADFS and CloudAccess. The purpose of these rules is to allow the user's email address and the role to pass through to SharePoint.

To add the claim rules for incoming assertions from CloudAccess:

1. Log in to your ADFS system.
2. Access the **Claims Provider Trusts** for CloudAccess.
3. Click **Edit Claim Rules**.
4. Add two rules 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 "Adding Claims Rules for Transforming Assertions for SharePoint" on page 111.

Adding Claims Rules for Transforming Assertions for SharePoint

You must configure ADFS to map the user's Email Address to Login on the SharePoint system, and to send the user's role.

To add claim rules for assertions sent to SharePoint:

1. In the ADFS console, select **Trust Relationships** > **Relying Party Trusts**.
2. Right-click *Name of your SharePoint system*, then select **Edit Claim Rules**.
3. Add 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 13.4.4, "Configuring the SharePoint People Picker to Use the Roles," on page 111.

13.4.4 Configuring the SharePoint People Picker to Use 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.

Before you begin, ensure that you have roles enabled within the SharePoint system using PowerShell scripts.

After you complete the ADFS configuration, you must configure the SharePoint option of **People Picker** to use the roles ADMIN and USER for claims received from ADFS.

1. Where the SharePoint system grants access, select **People Picker**.
2. Under **ADFS**, select **Role**.
3. In the **Find** box, 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, "Adding Roles to the SAML 2.0 Connector for ADFS," on page 108.
4. Select the role SharePoint returns, then assign the role to the group within SharePoint.

13.4.5 Troubleshooting SharePoint Issues

Use the following information if you encounter problems.

Issue: Error: The root of the certificate chain is not a trusted root authority.

Solution: You need to change the SharePoint server certificates. For detailed instructions, see *Root Certificate Chain not Trusted* (http://blogs.technet.com/b/speschka/archive/2010/02/13/root-of-certificate-chain-not-trusted-error-with-claims-authentication.aspx).

14 Connector for ADFS (WS-Federation)

The connector for Active Directory Federation Services (ADFS) provides federated single sign-on (SSO) access to ADFS with WS-Federation through CloudAccess. It does not support provisioning. The connector allows CloudAccess to authenticate a user against your identity sources and to share this authentication with ADFS in order to establish the user's session.

You can download the WS-Federation connector for ADFS from NetIQ Downloads (https://dl.netiq.com/). You must import the connector to CloudAccess, configure it to work with your ADFS system, and then map policies to set entitlements to ADFS for your users. You must also configure ADFS to work with the connector.

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

14.1 Requirements

Verify that you meet the following requirements before you import the connector:

❒ An understanding of 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 *Understanding WS-Federation* in the Microsoft Developer Network Library (http://msdn.microsoft.com/en-us/library/bb498017.aspx).

❒ An ADFS 2.0 system, installed and configured.

❒ Administrator access to the ADFS system. An understanding of ADFS and its management tools are presumed.

❒ An ADFS user account for each user who wants to authenticate to ADFS through the CloudAccess single sign-on service. The connector for ADFS does not provision user accounts.

❒ The location in the ADFS administration console where you will configure the WS-Federation federation for CloudAccess.

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

❒ The metadata file from the ADFS 2.0 system.

```
https://adfsserver/FederationMetadata/2007-06/FederationMetadata.xml
```

You will need the following information from the metadata file:

- **Login URL:** The value in the **PassiveRequestorEndpoint** field. For example:

  `https://`*`adfsserver`*`/adfs/ls/`

- **EntityID:** The value in the **entityID** field.

NOTE: ADFS does not provide a logout URL for WS-Federation. Users who log in to ADFS through CloudAccess must close their browser in order to log out.

14.2 Configuring the Connector

After you import the connector, you must configure it to work with your ADFS system. Perform the following task with a provider administrator account.

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

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

2. Drag and drop the WS-Federation connector for ADFS from the **Applications** palette to the **Applications** panel.

   The Configuration window opens automatically for the initial configuration. To view or reconfigure the settings later, click the connector icon, then click **Configure**.

3. On the **Configuration** page, specify the configuration properties.

   Use the information from the ADFS metadata file. The signing certificate from ADFS is optional.

4. Under **Assertion Attribute Mappings**, map the WS-Federation Assertion attributes to the attributes in your identity source.

5. Expand the **Federation Instructions**, then copy and paste the instructions into a text file to use during the ADFS 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.

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

   For more information, see Section 2.5, "Configuring Appmarks for Connectors," on page 29.

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 ADFS as the ADFS administrator, then configure the WS-Federation federation for CloudAccess in the ADFS 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** in the toolbar, then perform policy mapping to specify entitlements for identity source roles (groups).

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

12. Users can log in through CloudAccess to single sign-on to the ADFS system. The CloudAccess login page URL is:

```
https://appliance_dns_name
```

13 If you want users to connect to SharePoint, proceed to Section 14.4, “Connecting to SharePoint,” on page 115.

14.3 Troubleshooting Certificate Errors

If you use a self-signed certificate and certificate chain errors occur, run the following PowerShell command:

```
Set-ADFSClaimsProviderTrust -TargetName Display Name from above -
SigningCertificateRevocationCheck None
```

14.4 Connecting to SharePoint

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

This section describes how you can leverage the claims-based single sign-on capabilities of ADFS and SharePoint to set up a hub model of federation through ADFS. In this hub, CloudAccess has a trusted relationship with ADFS as the identity provider, and ADFS has a trusted relationship with SharePoint as a claims-based federation provider. SharePoint accepts the claims-based assertions, and allows users to access federated SharePoint web applications. Using roles for claim-based single sign-on makes it easier for SharePoint site administrators to map role and organization claims to SharePoint groups.

To set up the relationships, you define the roles in the connector for ADFS that ADFS and SharePoint will use for the claims-based single sign-on. The connector adds the role information to the identity information in assertions that it sends to ADFS.

In ADFS, you configure claims rules that look for the email address and role of users, and then transform them for use by SharePoint. ADFS applies rules to the assertions from CloudAccess to transform them into role claims that the SharePoint web applications understand, and sends the role claims to SharePoint.

In SharePoint, you configure its Person Picker to look for the roles in the assertions from ADFS. SharePoint validates the assertion information, stores the information in its token cache, and issues a session cookie for the user. By default, SharePoint sets the session lifetime to be the same as the token lifetime. In ADFS, you can specify the web single sign-on lifetime that determines the lifetime of the session cookie. Typically, the cookie expires when the user closes the browser window.

To set up this claims-based single sign-on federation hub:

- The CloudAccess administrator must modify the definition for the connector for ADFS to add two new roles to use for claims-based single sign-on, and then import and configure the modified connector.
- The ADFS administrator must configure a connection between SharePoint and ADFS, and define the rules for passing identity and role information from CloudAccess to SharePoint.
- The SharePoint administrator must modify the SharePoint People Picker to look for the roles in incoming assertions.
- The SharePoint administrator can add users to a SharePoint group based on the users' roles.

Use the following information to set up a federation relationship between SharePoint and CloudAccess that uses ADFS as a federation provider for SharePoint.

- Section 14.4.1, "Requirements," on page 116
- Section 14.4.2, "Adding Roles to the WS-Federation Connector for ADFS," on page 116
- Section 14.4.3, "Modifying Claims Rules in the ADFS System," on page 118
- Section 14.4.4, "Configuring the SharePoint People Picker to Use the Roles," on page 119
- Section 14.4.5, "Troubleshooting SharePoint Issues," on page 119

14.4.1 Requirements

Verify that you meet the following requirements:

- ❐ A CloudAccess appliance, installed and configured.
- ❐ One server with the following components installed:
  - ❐ Windows Server 2008 (or later) with the latest updates.
  - ❐ Active Directory with the latest updates.
  - ❐ ADFS 2.0 with the latest updates.
- ❐ SharePoint 2010 (or later) server with the latest updates, installed in the same domain as the ADFS server.
  - ❐ The SharePoint server should be connected to the ADFS server.

    For information about connecting the servers, see the following references in the Microsoft TechNet Library:

    - *How to Configure ADFS v 2.0 in SharePoint Server 2010* (http://technet.microsoft.com/en-us/library/hh305235%28v=office.14%29.aspx).
    - *Configure SAML-based Claims Authentication with ADFS in SharePoint 2013* (http://technet.microsoft.com/en-us/library/hh305235%28v=office.15%29.aspx)
  - ❐ Roles enabled within the SharePoint system using PowerShell scripts.

14.4.2 Adding Roles to the WS-Federation Connector for ADFS

You must modify the definitions in a WS-Federation connector for ADFS template file to add roles that will be used when ADFS sends role claims to SharePoint. These instructions create two roles: an administrator role called `ADMIN` and a user role called `USER`.

- "Modifying the WS-Federation Connector for ADFS Template" on page 116
- "Importing the Modified Connector" on page 117
- "Configuring the Modified Connector" on page 117

Modifying the WS-Federation Connector for ADFS Template

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

1. Obtain a copy of the ZIP file for the WS-Federation 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 **Pre-defined,** then select **Role**.

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

Name: Specify `ADMIN`.

Description: Specify `Administrator Role`.

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

Name: Specify `USER`.

Description: Specify `User Role`.

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

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 "Importing the Modified Connector" on page 117.

Importing the Modified Connector

After you modify the WS-Federation 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 WS-Federation connector for ADFS.

4 Click **Import**.

The **Applications** palette displays the modified WS-Federation connector for ADFS.

5 Proceed to "Configuring the Modified Connector" on page 117.

Configuring the Modified Connector

After you export and import the modified connector, you configure the connector by following the steps in Section 14.2, "Configuring the Connector," on page 114.

After you configure a WS-Federation connector for ADFS that supports SharePoint roles, you must modify ADFS and SharePoint to accept these roles. Proceed to Section 14.4.3, "Modifying Claims Rules in the ADFS System," on page 118.

14.4.3 Modifying Claims Rules in the ADFS System

Before you begin, ensure that you have configured a connection between ADFS and SharePoint. In ADFS, you must define the claim rules for incoming assertions from CloudAccess and for outgoing assertions sent to SharePoint.

- "Adding Claims Rules for SharePoint Roles in Incoming Assertions" on page 118
- "Adding Claims Rules for Transforming Assertions for SharePoint" on page 118

Adding Claims Rules for SharePoint Roles in Incoming Assertions

You must add ADFS claim rules between ADFS and CloudAccess. The purpose of these rules is to allow the user's email address and the role to pass through to SharePoint.

To add the claim rules for incoming assertions from CloudAccess:

1. Log in to your ADFS system.
2. Access the **Claims Provider Trusts** for CloudAccess.
3. Click **Edit Claim Rules**.
4. Add two rules 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 "Adding Claims Rules for Transforming Assertions for SharePoint" on page 118.

Adding Claims Rules for Transforming Assertions for SharePoint

You must configure ADFS to map the user's Email Address to Login on the SharePoint system, and to send the user's role.

To add the claim rules for assertions sent to SharePoint:

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. Add 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 14.4.4, "Configuring the SharePoint People Picker to Use the Roles," on page 119.

14.4.4 Configuring the SharePoint People Picker to Use 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 WS-Federation assertion.

Before you begin, ensure that you have roles enabled within the SharePoint system using PowerShell scripts.

After you complete the ADFS configuration, you must configure the SharePoint option of **People Picker** to use the roles ADMIN and USER for claims received from ADFS.

1. Where the SharePoint 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 14.4.2, "Adding Roles to the WS-Federation Connector for ADFS," on page 116.
4. Select the role SharePoint returns, then assign the role to the group within SharePoint.

14.4.5 Troubleshooting SharePoint Issues

Use the following information if you encounter problems.

Issue: Error: The root of the certificate chain is not a trusted root authority.

Solution: You need to change the SharePoint server certificates. For detailed instructions, see *Root Certificate Chain not Trusted* (http://blogs.technet.com/b/speschka/archive/2010/02/13/root-of-certificate-chain-not-trusted-error-with-claims-authentication.aspx).

15 Connector for Azure (WS-Federation)

The connector for Azure provides federated single sign-on (SSO) access to Azure with WS-Federation through CloudAccess. It does not support provisioning. The connector allows CloudAccess to authenticate a user against your identity sources and to share this authentication with Azure in order to establish the user's session.

You can download the connector for Azure from NetIQ Downloads (https://dl.netiq.com/). You must import the connector to CloudAccess, configure it to work with your Azure account, and then map policies to set entitlements to Azure for your users. You must also configure Azure to work with the connector.

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

- Section 15.1, "Requirements," on page 121
- Section 15.2, "Configuring the Connector," on page 122

15.1 Requirements

Verify that you meet the following requirements before you import the connector:

❐ An understanding of 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 *Understanding WS-Federation* in the Microsoft Developer Network Library (http://msdn.microsoft.com/en-us/library/bb498017.aspx).

❐ A Microsoft Azure account.

❐ A Windows Azure Access Control Service.

For more information, see *How to Authenticate Web Users with Windows Azure Access Control Service* (http://www.windowsazure.com/en-us/develop/net/how-to-guides/access-control/).

❐ Administrator access to the Azure account. An understanding of Azure and its account management tools are presumed.

❐ An Azure user account for each user who wants to authenticate to Azure through the CloudAccess single sign-on service. The connector for Azure does not provision user accounts.

❐ The location in the Azure administration console where you will configure the WS-Federation federation for CloudAccess.

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

❐ The metadata file from the Windows Azure Access Control Service.

Login URL: The value found in the **PassiveRequestorEndpoint** field. For example:

```
https://ncssacs.accesscontrol.windows.net/v2/WS-Federation
```

EntityID: The value in the **entityID** field.

NOTE: Azure does not provide a logout URL for WS-Federation. Users who log in to Azure through CloudAccess must close their browser in order to log out.

❐ If the connector for Office 365 is configured in the same domain, ensure that you configure WS-Federation as the federation protocol for the connector for Office 365.

15.2 Configuring the Connector

After you import the connector, you must configure it to work with the Azure system. Perform the following task with a provider administrator account.

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

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

2 Drag and drop the WS-Federation connector for Azure from the **Applications** palette to the **Applications** panel.

The Configuration window opens automatically for the initial configuration. To view or reconfigure the settings later, click the connector icon, then click **Configure**.

3 On the **Configuration** page, specify the configuration properties.

Use the information from the Azure metadata file.

4 Under **Assertion Attribute Mappings**, map the WS-Federation Assertion attributes to the attributes in your identity source.

5 Expand the **Federation Instructions**, then copy and paste the instructions into a text file to use during the Azure 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.

---

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

For more information, see Section 2.5, “Configuring Appmarks for Connectors,” on page 29.

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 Azure as the Azure administrator, then configure the WS-Federation federation for CloudAccess in the Azure 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** in the toolbar, then perform policy mapping to specify entitlements for identity source roles (groups).

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

12 After you complete the configuration, users can log in to the Azure application.

Users can also log in through CloudAccess to single sign-on to the Azure system. The CloudAccess login page URL is:

```
https://appliance_dns_name
```

16 Connector for Box (SAML 2.0)

The connector for Box provides federated single sign-on (SSO) access to Box with SAML 2.0 through CloudAccess. It does not support provisioning. The connector allows CloudAccess to authenticate a user against your identity sources and to share this authentication with Box in order to establish the user's session.

You can download the connector for Box from NetIQ Downloads (https://dl.netiq.com/). You must import the connector to CloudAccess, configure it to work with your Box account, and then map policies to set entitlements to Box for your users. You must also configure Box to work with the connector.

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

- Section 16.1, "Requirements," on page 125
- Section 16.2, "Configuring the Connector," on page 126

16.1 Requirements

Verify that you meet the following requirements before you import the connector:

- ❒ 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).
- ❒ An enterprise Box account.
- ❒ Administrator access to your enterprise Box account. An understanding of Box and its account management tools are presumed.
- ❒ A Box user account for each user who wants to authenticate to Box through the CloudAccess single sign-on service. The connector for Box does not provision user accounts.
- ❒ Box requires you to contact your Box account representative to configure the SAML 2.0 federation between Box and CloudAccess. Be prepared to provide them the federation configuration details for your organization at that time.

  When you configure the connector, the **Federation Instructions** provide the information that your Box account representative will need to set up the federation in Box for CloudAccess. This information includes the metadata; the field values to use; and other guidance.
- ❒ (Optional) An X.509 signing certificate from Box is required to support single logout. Communications use SSL regardless of whether you provide this certificate.

16.2 Configuring the Connector

After you import the connector, you must configure the connector settings in CloudAccess.

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

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

2 Drag and drop the connector for Box from the **Applications** palette to the **Applications** panel.

The Configuration window opens automatically for the initial configuration. To view or reconfigure the settings later, click the connector icon, then click **Configure**.

3 On the **Configuration** page, specify the configuration properties.

The signing certificate from Box is optional.

4 Under **Assertion Attribute Mappings**, map the SAML Assertion attributes to the appropriate attributes in your identity source.

5 Expand the **Federation Instructions**, then copy and paste the instructions into a text file to use during the Box 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.

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

For more information, see Section 2.5, "Configuring Appmarks for Connectors," on page 29.

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 Contact your Box account representative and provide the information they need to configure the SAML 2.0 federation for CloudAccess.

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 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 *NetIQ® CloudAccess and MobileAccess Installation and Configuration Guide*.

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

```
https://appliance_dns_name
```

17 Connector for Jive (SAML 2.0)

The connector for Jive provides federated single sign-on (SSO) access to Jive with SAML 2.0 through CloudAccess. It does not support provisioning. The connector allows CloudAccess to authenticate a user against your identity sources and to share this authentication with Jive in order to establish the user's session.

You can download the connector for Jive from NetIQ Downloads (https://dl.netiq.com/). You must import the connector to CloudAccess, configure it to work with your Jive account, and then map policies to set entitlements to Jive for your users. You must also configure Jive to work with the connector.

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

17.1 Requirements

Verify that you meet the following requirements before you import the connector:

- ☐ 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 Jive account.
- ☐ Administrator access to your Jive account. An understanding of Jive and its account management tools are presumed.
- ☐ A Jive user account for each user who wants to authenticate to Jive through the CloudAccess single sign-on service. The connector for Jive does not provision user accounts.
- ☐ The location in the Jive administration console where you will configure the SAML 2.0 federation for CloudAccess.

  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, Jive requires you to contact Jive Technical Support to configure the SAML 2.0 federation between Jive and CloudAccess. Be prepared to provide them the federation configuration details for your organization at that time.

  When you configure the connector, the **Federation Instructions** provide the information that you (or Jive Technical Support) will need to set up the federation in Jive for CloudAccess. This information includes the metadata; a signing certificate for the appliance; the field values to use; and other guidance.
- ☐ If you are using a Jive Hosted account or a Jive On Prem account, the Base Metadata URL that is displayed on the SAML 2.0 federation pages on Jive.

❒ If you are using a Jive Cloud account, the Instance name of your account.

❒ (Optional) An X.509 signing certificate from Jive is required to support single logout. Communications use SSL regardless of whether you provide this certificate.

❒ CloudAccess supports one online instance and one offline instance of Jive. Each instance must have a unique Base URL and SAML assertion.

17.2 Configuring the Connector

After you import the connector, you must configure the connector to work with Jive.

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

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

2 Drag and drop the connector for Jive from the **Applications** palette to the **Applications** panel.

The Configuration window opens automatically for the initial configuration. To view or reconfigure the settings later, click the connector icon, then click **Configure**.

3 On the **Configuration** page, specify the configuration properties.

Use the Base Metadata URL or the Instance name for the Jive account. The signing certificate from Jive is optional.

4 Under **Assertion Attribute Mappings**, map the SAML Assertion attributes to the appropriate attributes in your identity source.

5 Expand the **Federation Instructions**, then copy and paste the instructions into a text file to use during the Jive 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.

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

For more information, see Section 2.5, “Configuring Appmarks for Connectors,” on page 29.

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 Do one of the following:

- If you are using a Jive Hosted account or a Jive On Prem account, log in to Jive as the account administrator, then configure the SAML 2.0 federation for CloudAccess in the Jive administration console.
- If you are using a Jive Cloud account, contact Jive Technical Support and provide the information they need to configure the SAML 2.0 federation for CloudAccess.

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

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

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

```
https://appliance_dns_name
```

18 Connector for ServiceNow (SAML 2.0)

The connector for ServiceNow provides federated single sign-on (SSO) access to ServiceNow with SAML 2.0 through CloudAccess. It does not support provisioning. The connector allows CloudAccess to authenticate a user against your identity sources and to share this authentication with ServiceNow in order to establish the user's session.

You can download the connector for ServiceNow from NetIQ Downloads (https://dl.netiq.com/). You must import the connector to CloudAccess, configure it to work with your ServiceNow account, and then map policies to set entitlements to ServiceNow for your users. You must also configure ServiceNow to work with the connector.

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

- Section 18.1, "Requirements," on page 131
- Section 18.2, "Configuring the Connector," on page 132

18.1 Requirements

Verify that you meet the following requirements before you import the connector:

❒ 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 management ServiceNow account, created with the SAML 2.0 Update 1 plug-in.

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

❒ A ServiceNow user account for each user who wants to authenticate to ServiceNow through the CloudAccess single sign-on service. The connector for ServiceNow does not provision user accounts.

❒ 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; a signing certificate for the appliance; the field values to use; and other guidance.

❒ The Instance name for your ServiceNow account. For example:

```
http://your_instance.service-now.com/
```

❒ (Optional) An X.509 signing certificate from ServiceNow is required to support single logout. Communications use SSL regardless of whether you provide this certificate.

18.2 Configuring the Connector

After you import the connector, you must configure the connector to work with ServiceNow.

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

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

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

   The Configuration window opens automatically for the initial configuration. To view or reconfigure the settings later, click the connector icon, then click **Configure**.

3 On the **Configuration** page, specify the configuration properties.

   Use the instance name of your ServiceNow account. The signing certificate from ServiceNow is optional.

4 Under **Assertion Attribute Mappings**, map the SAML Assertion attributes to the appropriate attributes in your identity source.

5 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.

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

   For more information, see Section 2.5, "Configuring Appmarks for Connectors," on page 29.

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 ServiceNow as the ServiceNow administrator, then configure the SAML 2.0 federation for CloudAccess in the ServiceNow 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** in the toolbar, then perform policy mapping to specify entitlements for identity source roles (groups).

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

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

   ```
   https://appliance_dns_name
   ```

19 Connector for VMware vCloud (SAML 2.0)

The connector for VMware vCloud provides federated single sign-on (SSO) capabilities to vCloud through CloudAccess. It does not support provisioning. The connector allows CloudAccess to authenticate a user against your identity sources and to share this authentication with vCloud in order to establish the user's session.

You can download the connector for vCloud from NetIQ Downloads (https://dl.netiq.com/). You must import the connector to CloudAccess, configure it to work with vCloud, and then map policies to set entitlements to vCloud for your users. You must also configure vCloud to work with the connector.

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

19.1 Requirements

Verify that you meet the following requirements before you import the connector:

❒ 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 VMware vCloud deployment with a vCloud director.

❒ Administrator access to the vCloud system. An understanding of vCloud and its management tools are presumed.

❒ A vCloud user account for each user who wants to authenticate to vCloud through the CloudAccess single sign-on service. The connector for VMware vCloud does not provision user accounts.

❒ The location in the vCloud 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 vCloud for CloudAccess. This information includes the metadata; a signing certificate for the appliance; the field values to use; and other guidance.

❒ You will need the following information:

Destination URL (optional): The URL that vCloud displays when a user logs in to vCloud.

vCloud Host: The IP address or DNS name of the vCloud Director.

Organization: The name of your vCloud organization.

❒ (Optional) An X.509 signing certificate from vCloud is required to support single logout. Communications use SSL regardless of whether you provide this certificate.

19.2 Configuring the Connector

After you import the connector, you must configure the connector to work with vCloud.

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

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

2 Drag and drop the connector for VMware vCloud from the **Applications** palette to the **Applications** panel.

The Configuration window opens automatically for the initial configuration. To view or reconfigure the settings later, click the connector icon, then click **Configure**.

3 On the **Configuration** page, specify the configuration properties.

Use the information from your vCloud system. The signing certificate from vCloud is optional.

4 Under **Assertion Attribute Mappings**, map the SAML Assertion attributes to the appropriate attributes in your identity source.

5 Expand the **Federation Instructions**, then copy and paste the instructions into a text file to use during the vCloud 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.

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

For more information, see Section 2.5, "Configuring Appmarks for Connectors," on page 29.

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

8 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.

9 Log in to the vCloud director as the vCloud administrator, then configure the SAML 2.0 federation for CloudAccess in the vCloud 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.

10 In the CloudAccess administration console, 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 *NetIQ® CloudAccess and MobileAccess Installation and Configuration Guide*.

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

```
https://appliance_dns_name
```

20 Connector for WebEx (SAML 2.0)

The connector for WebEx provides federated single sign-on (SSO) access to WebEx with SAML 2.0 through CloudAccess. It does not support provisioning. The connector allows CloudAccess to authenticate a user against your identity sources and to share this authentication with WebEx in order to establish the user's session.

You can download the connector for WebEx from NetIQ Downloads (https://dl.netiq.com/). You must import the connector to CloudAccess, configure it to work with your WebEx account, and then map policies to set entitlements to WebEx for your users. You must also configure WebEx to work with the connector.

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

- Section 20.1, "Requirements," on page 135
- Section 20.2, "Configuring the Connector," on page 136

20.1 Requirements

Verify that you meet the following requirements before you import the connector:

❒ 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 WebEx account. Trial accounts do not support federation.

❒ Administrator access to your WebEx account. An understanding of WebEx and its account management tools are presumed.

❒ WebEx user accounts for each user who wants access to the single sign-on service. The connector for WebEx does not provision user accounts.

❒ The location in the WebEx 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 WebEx for CloudAccess. This information includes the metadata; the field values to use; and other guidance.

❒ The domain name for your WebEx account. For example:

```
https://custom-ID.webex.com
```

❒ (Optional) An X.509 signing certificate from WebEx is required to support single logout. Communications use SSL regardless of whether you provide this certificate.

20.2 Configuring the Connector

After you import the connector, you must configure the connector to work with your WebEx system. To configure the connector for WebEx:

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

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

2 Drag and drop the SAML 2.0 connector for WebEx from the **Applications** palette to the **Applications** panel.

The Configuration window opens automatically for the initial configuration. To view or reconfigure the settings later, click the connector icon, then click **Configure**.

3 On the **Configuration** page, specify the configuration properties.

Use the domain name of your WebEx account. The signing certificate from WebEx is optional.

4 Under **Assertion Attribute Mappings**, map the SAML Assertion attributes to the appropriate attributes in your identity source.

5 Expand the **Federation Instructions**, then copy and paste the instructions into a text file to use during the WebEx 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.

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

For more information, see Section 2.5, "Configuring Appmarks for Connectors," on page 29.

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 WebEx as the WebEx administrator, then configure the SAML 2.0 federation for CloudAccess in the WebEx 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** in the toolbar, then perform policy mapping to specify entitlements for identity source roles (groups).

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

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

```
https://appliance_dns_name
```

21 Connector for Zoho (SAML 2.0)

The connector for Zoho provides federated single sign-on (SSO) access to Zoho with SAML 2.0 through CloudAccess. It does not support provisioning. The connector allows CloudAccess to authenticate a user against your identity sources and to share this authentication with Zoho in order to establish the user's session.

You can download the connector for Zoho from NetIQ Downloads (https://dl.netiq.com/). You must import the connector to CloudAccess, configure it to work with your Zoho account, and then map policies to set entitlements to Zoho for your users. You must also configure Zoho to work with the connector.

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

- Section 21.1, "Requirements," on page 137
- Section 21.2, "Configuring the Connector," on page 138

21.1 Requirements

Verify that you meet the following requirements before you import the connector:

- ❒ 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 business Zoho account.
- ❒ Administrator access to your Zoho account. An understanding of Zoho and its account management tools are presumed.
- ❒ A Zoho user account for each user who wants to authenticate to Zoho through the CloudAccess single sign-on service. The connector for Zoho does not provision user accounts.
- ❒ The location in the Zoho 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 Zoho for CloudAccess. This information includes the metadata; the field values to use; and other guidance.
- ❒ A valid public domain that you registered with Zoho when you created your Zoho account. Select the **Enable MailHosting** option after logging in to the Zoho account.
- ❒ (Optional) An X.509 signing certificate from Zoho is required to support single logout. Communications use SSL regardless of whether you provide this certificate.

21.2 Configuring the Connector

After you import the connector, you must configure the connector to work with Zoho.

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

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

2 Drag and drop the SAML 2.0 connector for Zoho from the **Applications** palette to the **Applications** panel.

   The Configuration window opens automatically for the initial configuration. To view or reconfigure the settings later, click the connector icon, then click **Configure**.

3 On the **Configuration** page, specify the configuration properties.

   Use the domain name that you registered for your Zoho account. The signing certificate from Zoho is optional.

4 Under **Assertion Attribute Mappings**, map the SAML Assertion attributes to the appropriate attributes in your identity source.

5 Expand the **Federation Instructions**, then copy and paste the instructions into a text file to use during the Zoho 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.

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

   For more information, see Section 2.5, “Configuring Appmarks for Connectors,” on page 29.

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 Zoho as the Zoho administrator, then configure the SAML 2.0 federation for CloudAccess in the Zoho 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** in the toolbar, then perform policy mapping to specify entitlements for identity source roles (groups).

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

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

```
https://appliance_dns_name
```

22 Troubleshooting CloudAccess

Use the information in the following sections to troubleshoot any issues you might encounter.

22.1 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 22-1 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 22-1 *Troubleshooting Log Files for Application Access Issues*

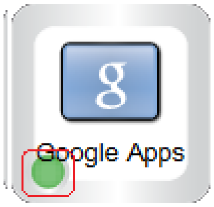
Feature	Logs
Identity Source Provisioning	`bis_AD_<xxxxx>.log`
	`bis_AD_<xxxxx>_RL.log`
	`ConnectorLogs.txt`
	`bis_EDIR_h2q3p.log`
	`bis_EDIR_h2q3p_RL.log`
Provisioning to the SaaS Applications	`connectors_SFORCE_<xxxxx>_RL.log`
	`connectors_GOOGLEAPPS_<xxxxx>.log`
	`connectors_GOOGLEAPPS_<xxxxx>_RL.log`
	`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`

22.2 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 22-1 *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.

22.3 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, and Office 365). The following table lists the actions in the identity sources and the corresponding actions in the SaaS applications.

Table 22-2 *Provisioning Actions*

Identity Sources	SaaS Applications
Delete a user. (Or disable the user account.)	Disables the SaaS account. **NOTE:** In the MobileAccess app 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 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.

22.4 Troubleshooting Google Apps Issues

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.

22.5 Troubleshooting Salesforce 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.

22.6 Troubleshooting Office 365 Issues

Use the information in the following sections to help you troubleshoot issues with the connector for Office 365:

- Section 22.6.1, "Obtaining Installation and Provisioning Logs," on page 143
- Section 22.6.2, "Office 365 Logout Error on Mobile Devices," on page 143

22.6.1 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 5.3, "Installing the Connector for Office 365," on page 63.

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).

22.6.2 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**.

22.7 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 NetIQ 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 NetIQ Technical Support (NTS) (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.

Before you contact NetIQ Technical Support, please complete the appropriate worksheet for the connector type that you want to create. The more information that you can provide, the better and quicker NTS can help you create the connector.

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 ❐ SAML 1.*x*
□	Is the metadata (SAML/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?

□	**Gather the following information:**
□	What user information is required by your partner for SSO? For example: email address, CN, and so on.
□	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.

□ **Gather the following information:**

□ What user information does the partner send for SSO? For example: email address, CN, and so on.

□ 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 a destination website. 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* in 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)

□ What is the message for a successful login?

After a successful login, what word or phrase appears on the default web page that opens for any user?

□ What is the message for a failed login?

□ On what domain is the form?