CA CloudMinder™

Getting Started with SSO

1.51

Contact CA Technologies

Contact CA Support

For your convenience, CA Technologies provides one site where you can access the information that you need for your Home Office, Small Business, and Enterprise CA Technologies products. At http://ca.com/support, you can access the following resources:

- Online and telephone contact information for technical assistance and customer services
- Information about user communities and forums
- Product and documentation downloads
- CA Support policies and guidelines
- Other helpful resources appropriate for your product

Providing Feedback About Product Documentation

If you have comments or questions about CA Technologies product documentation, you can send a message to techpubs@ca.com.

To provide feedback about CA Technologies product documentation, complete our short customer survey which is available on the CA Support website at http://ca.com/docs.

CA Technologies Product References

This document references the following CA Technologies products:

- CA CloudMinder™ Identity Management
- CA CloudMinder™ Advanced Authentication
- CA CloudMinder™ Single Sign-On
- CA Directory
- CA IdentityMinder™
- CA AuthMinder™
- CA RiskMinder™
- CA SiteMinder®
- CA SiteMinder® for Secure Proxy Server
- CA Layer 7

Contents

Chapter 1: SSO Service Tasks and Features

The intended audience of this guide is tenant administrators. The purpose is to make tenant administrators aware of the types of SSO services available through CA CloudMinder.

CA CloudMinder™ Single Sign-On (SSO) provides a cloud-based federation hub that lets customers connect to cloud-based applications, partner hosted applications or other on-premise applications in an organization.

The SSO service is standards-based. The service uses SAML, WS-Federation, and WS-Trust to securely share user identity information across business partners. Users log in one time and gain secure access to federated partner services and application without the inconvenience of maintaining different access credentials. Federated partnerships are deployed and maintained in the cloud so that your organization does not have to develop the infrastructure internally.

All types of users can single sign-on to a particular site. Users can be enterprise customers, such as employees that work at your company facility or from outside your corporate facility. Enterprise users can also be third-party partners that are not part of your organization. Users can be consumers that enroll in services that a company are offers, such as a rewards program. The SSO service can provide access to applications for these types of customers.

The following sections describe SSO functionality available for your business.

SSO Applications Configured for your Portal

Before a user can access an SSO application, add the application to your portal.

Applications that you add to the portal can be third-party applications or on-premise applications. The applications can serve enterprise customers, such as employees or business partners, or users requesting access to a specific consumer application.

From a configuration perspective, add an SSO application from the User Console and protect each application with an authentication method. The authentication method corresponds to an authentication scheme and a federated partnership that facilitates the SSO transaction. The hosting administrator configures the authentication scheme and partnership in the Cloud Service Provider (CSP) Console.

Review the instructions for creating an application (see page 58).

Authentication Methods for SSO Applications

An authentication method determines how a user authenticates when they request an SSO application. An authentication method must be associated with every SSO application.

A one-to-one correspondence exists between each *authentication method* that you associate with an application and an *authentication scheme* that is set up by the hosting administrator. As the tenant administrator, you are responsible for configuring the authentication method. The hosting administrator is responsible for configuring the authentication scheme and the SSO partnership.

The available authentication methods include:

Basic

Select this method when the SSO service authenticates the user. This authentication method is typically associated with the HTML form authentication scheme.

External IdP

Select this method when third-party business partners authenticate the user. The types of authentication schemes include federation protocols SAML 1.1, SAML 2.0, and WS-Federation. The schemes also include profiles that are typically used with social media sites, such as Google and Facebook. These schemes include OpenID and OAuth.

To obtain credentials, the SSO service authenticates the user directly or presents the user with a list of third-party sites. These external sites can serve as the identity provider and can authenticate the user. After the user authenticates successfully, the external IdP returns the user to the SSO service, which completes the transaction to the target application.

This option is only available with the SSO Service.

Advanced Authentication

Select this method for strong authentication that the Advanced Authentication Service provides. This service offers the following authentication methods for applications: Arcot OTP, Arcot PKI, Arcot OTP with Risk and Arcot PKI with Risk.

These options are only available with the Advanced Authentication Service.

Federated Partnerships to Enable SSO

The hosting administrator configures federated partnerships between your organization and your business partners to facilitate SSO transactions. The deployment determines the types of partnerships that are required, the SAML or WS-Federation profiles, and the attributes added to assertions.

When the hosting administrator sets up a partnership, the administrator can require information from you. If the hosting administrator is setting up a partnership with a well-known site, such as Facebook or Google, the required setup information can be minimal. The hosting administrator determines the appropriate protocols and authentication schemes for the deployment.

For a less well-known partner, the hosting administrator can require more information, such as:

- Federation protocol in use for SSO communication.
- Authentication method to collect proper credentials.
- Encryption algorithm that is required.
- User attributes that you require in the assertion.

Work together with the hosting administrator so that the deployment satisfies your requirements and the necessary information is available.

WS-Trust Claims Retrieval and Transformation

As an Identity Provider, the SSO service supports the transformation of claims in an assertion through a Security Token Service (STS). Security tokens can carry claims, which are attributes about a user. The ability to change claims through a Security Token Service enables SSO to endpoints with varied applications.

The STS is a third-party service that acts as the bridge between the IdP and the site with the target resources. The SSO service, acting as the IdP, can perform the following actions:

- Transform claims from the inbound claim to a corresponding outbound claim.
- Add claims to an assertion.
- Delete claims from an assertion.

As a tenant administrator, you can request the following services:

- STS access for a particular STS client. The SSO service can support claims transformation for SAML 1.1, SAML 2.0, and WS-Federation.
- Addition of a WS-Trust application to the user console.

Self-registration Services for SSO

A user who does not have an account at the tenant can be prompted to register during a single sign-on transaction. You can ask the hosting administrator to set up a partnership where an external service, such as Google or Facebook serves as the IdP. The user has to have an account with the external IdP to authenticate.

When a user requests a protected resource, they are redirected to an external IdP for authentication. After successful authentication, the IdP sends the user back to the SSO service where the user is prompted to register. Registration is optional.

The user is not limited to one external IdP. However, selecting a different external IdP for a subsequent request requires that they self-register again as a new user with a new account.

During the self-registration process, a user can set their password. If the user sets the password, the user can then log in to the application directly with a user name and password. If the user does not set the password, the user always has to log in with the external IdP first.

User Validation for Sensitive Tasks

User validation forces the user to reenter credentials for certain sensitive tasks. The goal of this feature is to prevent a different person from using an unattended browser to gain access to information. A user can open a browser session and can leave the browser unattended, or can forget to close all browser sessions. The session is now open for an unauthorized user to gain access to resources and perform tasks on those resources.

User validation confirms that the system validates that the end user matches the logged-in session. The system is not simply verifying that the client is valid.

User validation can be configured for user or administrative tasks, such as changing passwords. The feature provides an audit record for each verification, and it preserves the existing user session and session store contents.

If the session level of the user is equal or greater than the protection level of the resource, the user is not rechallenged.

You can ask the hosting administrator to configure user validation for your sensitive resources and tasks.

Attribute Query Support

The SSO service can serve as the IdP in an SSO transaction. The service can respond to attribute queries from an SP. The SP evaluates the additional attributes before granting access to the resource.

The attribute query feature works in two modes:

- SAML-compliant functionality

  Metadata lists all attributes for which a query response can be generated. Responding to specific attribute queries avoids sending infrequently used attributes.

- Extended functionality

  The SSO service accepts queries for attributes not listed in the metadata. The service checks the user directories first and then checks the session store for attributes. The session store contains dynamic attributes from the advanced authentication methods. The session store also contains dynamic proxied attributes from external IdPs.

Proxied Attribute Query Support

Proxied attribute query support is an addition to the standard attribute query support. This feature extends the search for attributes by passing queries to external IdPs.

The search for attributes proceeds in the following order:

1. User directories.
2. Session store.
3. External IdP, only if the attribute is not found in the user directory or session store, and if the user was initially authenticated by an external SAML 2.0 IdP.

   The SSO service queries the external IdP. If the external IdP finds the attribute, it responds to the SSO service with a query response. The SSO service adds the attributes from the external IdP to the session store. The SSO service returns the response with the attributes to the attribute requestor.

A hosting administrator can enable the proxied attribute query feature on a per-partnership basis.

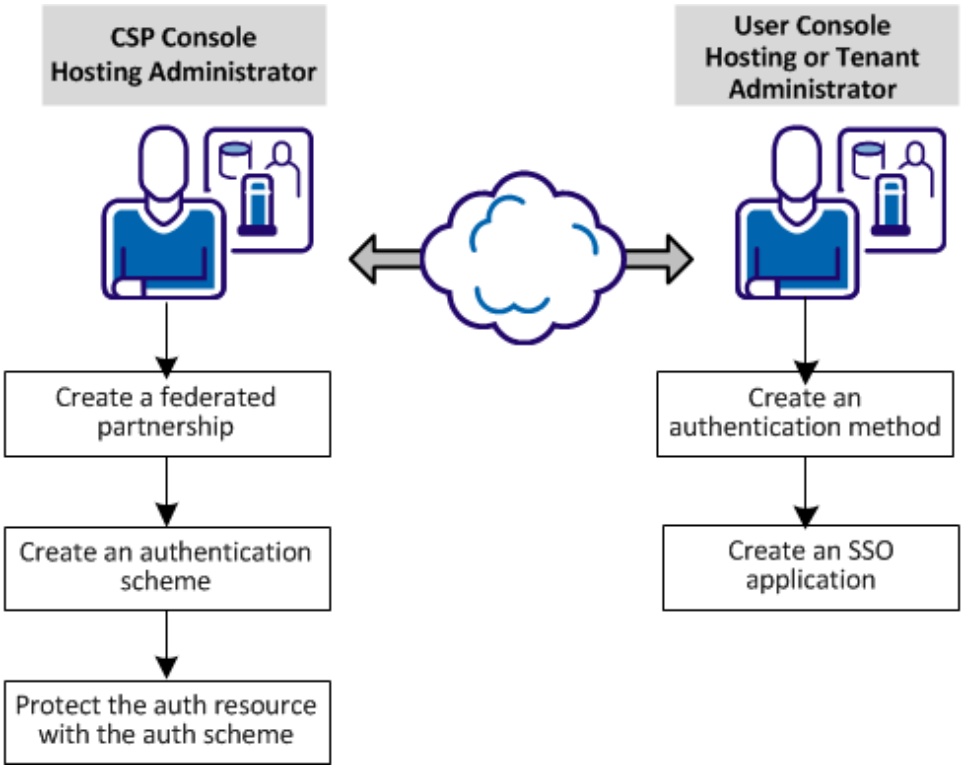
SSO Configuration Overview

Setting up single sign-on (SSO) for an application involves configuration tasks at the User Console and Administrative UI.

Prerequisites:

- Administrative UI is deployed.
- The tenant environment is deployed.
- The User Console is deployed.
- Third-party application is configured for SSO.

As a tenant administrator, your configuration tasks are performed at the User Console. The intent of the following diagram is to show tasks for the hosting and tenant administrators. All these tasks are necessary to complete single sign-on configuration.



At the Administrative UI:

1. Create an authentication scheme.
2. Protect the authentication resource with the authentication scheme.
3. Create a federated partnership.

At the User console:

4. Create authentication methods for the tenant.
5. Create an SSO application.

These configuration tasks are described in the scenario How to Set up Single Sign-on for an Application.

Chapter 2: SSO Using Advanced Authentication and Provisioning for a Sensitive Application

As an administrator, you want internal CA CloudMinder Identity Management users, such as employees or partners, to have easy but secure access to software resources outside of your network environment.

For example, Salesforce.com is a software resource outside of your network environment. You want all new sales employees in your company to have access to Salesforce. You want that access to be protected by security more advanced than the security provided by Salesforce. You want to set up Salesforce accounts for employees automatically, rather than creating them one-by-one. You also want your employees to be able to access Salesforce through single sign-on for enhanced convenience and security.

This scenario describes how to use CA CloudMinder to perform all of these activities:

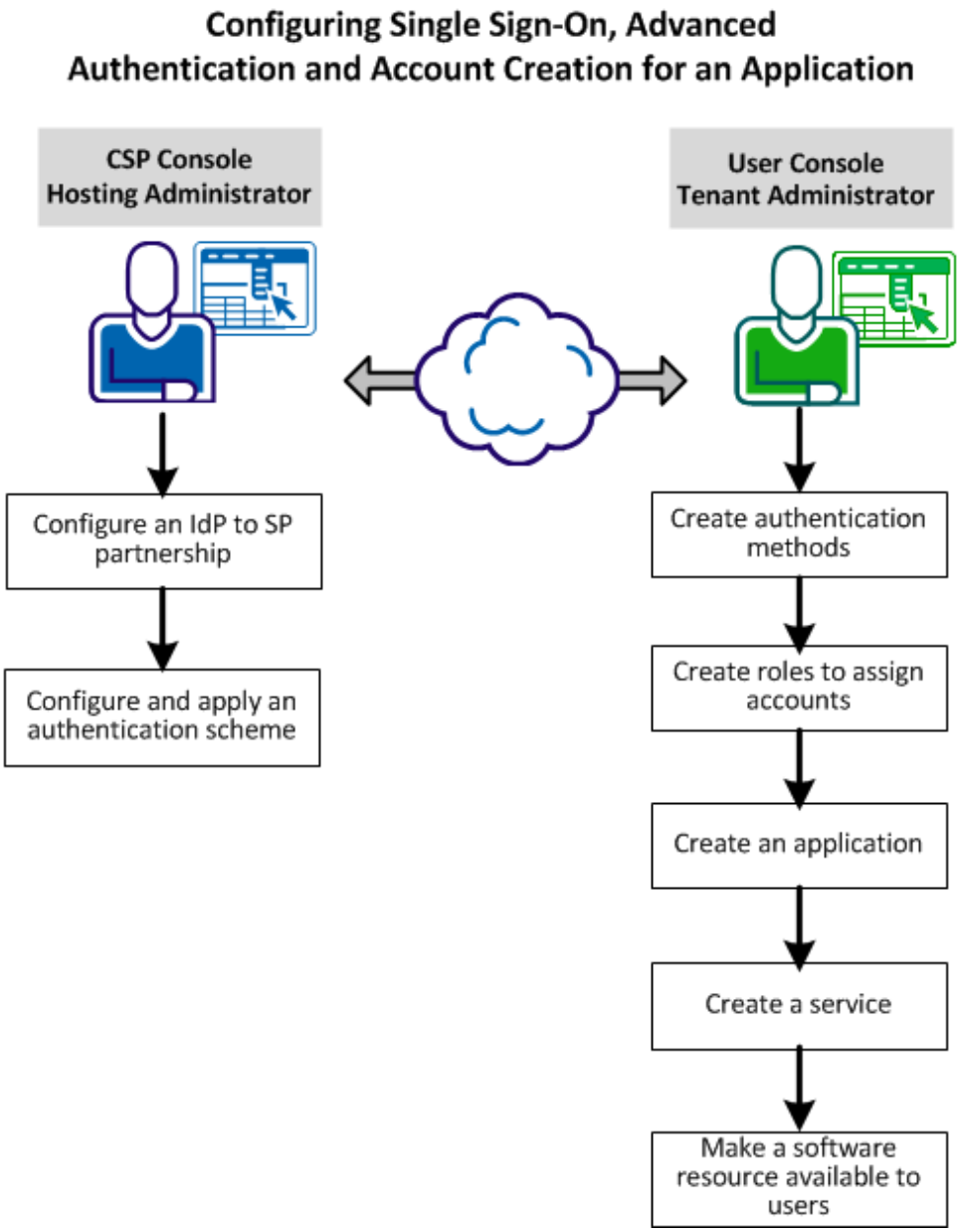
- configure a software resource for SSO access
- configure and implement advanced authentication
- configure automatic account creation (*provisioning)*

This scenario assumes that you have purchased the Single Sign-on Service, the Advanced Authentication Service, and the Provisioning Service for your CA CloudMinder environment.

Note: The terminology used for the target software resource is different depending on the component you are configuring. Note the following:

- In the Partnership section, the software resource is named a *resource*.
- In the Provisioning section, the software resource is named an *endpoint*.
- In the Authentication Scheme, Authentication Method, Application, and Service Wizard sections, the software resource is named an *application*.

The following figure shows the steps required to configure this scenario. In a typical environment, hosting administrators and tenant administrators have access to different system components and features. In this scenario, a hosting administrator performs some steps while a tenant administrator performs other steps.



Perform the following procedures to configure an application for SSO, advanced authentication, and account creation. (The responsible administrator is indicated in parenthesis.)

1. Configure an IdP to SP partnership (see page 18) (hosting administrator)

   Create federated partnerships to enable secure communication between your CA CloudMinder system and the target software resource.

2. Configure and apply an authentication scheme (see page 46) (hosting administrator)

   Configure and apply an authentication scheme to make a given type of login security available in your system. The authentication method and authentication scheme work together to protect access to the specified application.

3. Create authentication methods (see page 49) (tenant administrator)

   Create authentication methods to make a given type of login security available to apply to an application. The authentication method and authentication scheme work together to protect access to the specified application.

4. Create roles to assign accounts (see page 53) (tenant administrator)

   Create one or more account templates and provisioning roles to automatically create user accounts in the target software resource.

5. Create an application (see page 58) (tenant administrator)

   Create an application to define how and where users access the target software resource.

6. Create a service (see page 64) (tenant administrator)

   Create a service to give the user access to the application. The application is now configured with single sign-on, advanced authentication and automatic account creation.

7. Make the software resource available to users (see page 68) (tenant administrator)

   Make the resource available through the User Console, or through a URL link.

Configure an IdP to SP Partnership

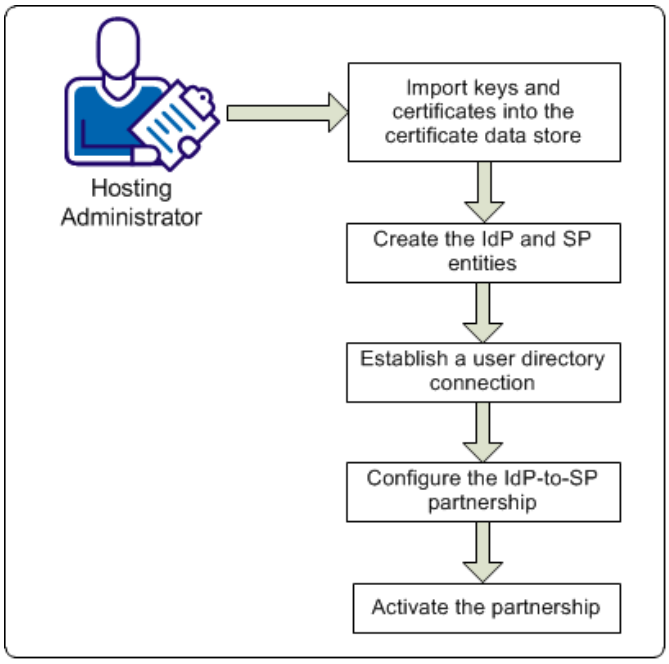
This scenario explains how to enable SSO to a sensitive software application owned by a partner, such as Salesforce.com. The application requires strong authentication due to confidential nature of the information. Also, you want to provision this user for the application after single sign-on is successful.

Set up an IdP-to-SP partnership between CA CloudMinder (IdP) and the business partner (SP) to enable this deployment.

In these instructions, the following information regarding SSO applies:

- The business partner is the Service Provider (SP).
- CA CloudMinder is the Identity Provider (IdP).
- The internal cloud user store is in use.
- SAML 2.0 is the federation profile in use.
- Identity Management provisions the user.

The following figure shows the configuration tasks required for IdP-to-SP partnership:



The following procedures explain how to set up the IdP-to-SP partnership:

1. Import keys and certificates into the certificate data store (see page 20).
2. Create the IdP and SP entities (see page 26).
3. Establish a user directory connection (see page 34).
4. Configure the IdP-to-SP partnership (see page 34).
5. Activate the partnership (see page 45).

Import Keys and Certificates into the Certificate Data Store

Private keys and certificates are required for the following tasks:

- Federation components use private key/certificate pairs for signing, verification, encryption, and decryption of entire assertions, or specific assertion content.
- Federation components employ client certificates for back-channel authentication for artifact single sign-on.
- For establishing SSL connections (SSL server certificates).

Private key/certificate pairs and single certificates for federation functions are stored in the certificate data store (CDS). The certificate data store is collocated with the policy store. All Policy Servers that share a common view into the same policy store have access to the same keys, certificates, CDS-configured certificate revocation lists (CRL), and OCSP responders.

SSL server certificates are stored on the web server where they are installed. SSL server certificates are not stored in the certificate data store.

Each key/certificate pair, client certificate, and trusted certificate in the certificate data store must have a unique alias. The alias allows any private key/certificate pair or single certificate in the certificate store can to be uniquely referenced. The certificate data store can store multiple key/certificate pairs and single certificates. In a federated environment, you can have multiple partners. For multiple partners, you can use a different pair for each partner.

If a signing alias is configured for signing assertions, the assertion generator uses the key that is associated with that alias to sign assertions. If no signing alias is configured, the assertion generator uses the key with the *defaultenterpriseprivatekey* alias to sign assertions. If the assertion generator does not find a default enterprise private key, it uses the first private key it finds to sign assertions.

Important! If you are going to store multiple keys, define the first key that you add with the defaultenterpriseprivatekey alias before adding subsequent keys.

A given Policy Server can sign or sign and verify responses. You can add keys and certificates for signing and validation to the same certificate data store.

You manage the contents of the certificate data store using the Administrative UI.

The following types of key/certificate pairs and single certificates are stored in the certificate data store:

Function	Private Key/Cert Pair	Certificate (public key)	CA Certificates	Client Certificate
Signs assertions, authentication requests, SLO requests and responses	X			
Verifies signed assertions, authentication requests, and SLO requests/responses		X		
Encrypts assertions, Name ID and attributes (SAML 2.0 only)		X		
Decrypts assertions, Name ID and attributes (SAML 2.0)	X			
Serves as a credential for client certificate authentication of the artifact back channel				X
Validates other certificates and certificate revocation lists			X	
Use SSL connections to resolve web services variables			X	

If you do not have a key/certificate pair in the certificate data store, you have two options:

- Import a key/certificate pair from an existing file (.p12 or .pfx).
- Generate a key/certificate pair.

  To generate a new key/certificate pair, request a certificate from a trusted Certificate Authority and then import the signed certificate response that the authority returns.

For more information about key and certificate management, see the *CA SiteMinder Policy Server Configuration Guide.*

Import a Key/Certificate Pair from an Existing File

If you do not have a key/certificate pair in the certificate data store, import one from an existing .p12 or .pfx file.

The Policy Server treats an imported certificate as a trusted certificate. The exceptions are self-signed certificates, which get treated according to the following guidelines:

- The Policy Server identifies a V3 self-signed certificate as a CA certificate. In this case, it treats it as a CA certificate. This behavior occurs even though you initiate the import from the Certificate/Private Key dialog.
- The Policy Server treats the certificate as a trusted certificate when:
  - The Policy Server does not identify a V3 self–signed certificate as a CA.
  - The certificate is a V1 self–signed certificate.

Follow these steps:

1. Log in to the Administrative UI.
2. Select Infrastructure, X509 Certificate Management, Trusted Certificates and Private Keys.
3. Click Import New and follow the wizard.
4. Be aware of the following items as you complete the wizard:
   - You can import a single file with a key and certificate in it or separate key and certificate files. Select the appropriate option button for the file you are using.
   - To import a self-signed certificate as a Certificate Authority certificate, set the Use as CA option button to Yes. The certificate is imported as a CA certificate and is not available for when configuring partnerships (for example, for signing or encryption).

     Otherwise, accept the default No setting to import the certificate as a trusted certificate that is available when configuring partnerships.
   - For a trusted certificate file in DER (binary) format, the file can contain one or more certificate entries. For a trusted certificate file in PEM (base 64) format, one certificate per file is required.
   - If you are using a .p12 file, you are required to fill in a password.
   - For each entry you plan to add to the certificate data store, enter the alias you want to associate with that entry. If you select multiple entries, each requires a unique alias.
5. At the Confirm step, review the information and click Finish.

The key/certificate pair is imported into the certificate data store.

How to Generate a Key/Certificate Pair

If you do not have a key/certificate pair in the certificate data store, you can generate a new key/certificate pair.

Follow these steps:

1. Generate a certificate request and send the request to a trusted Certificate Authority.
2. Import the signed certificate response from the authority.

Generate a Certificate Request

If you do not have a key/certificate pair in the certificate data store, request one from a trusted Certificate Authority. When the CA returns a signed certificate response, import it into the certificate data store.

When you generate a certificate request, the Policy Server generates a private key and a self–signed certificate pair. The Policy Server stores this pair in the certificate data store. Using the generated request, contact a Certificate Authority and fill out the CA certificate request form. Paste the contents of the generated request into the form.

The CA issues a signed certificate response, usually in PKCS #7 format. You can import the signed certificate response into the certificate data store. After the signed certificate response is imported, the existing self–signed certificate entry of the same alias is replaced.

Follow these steps:

1. Log in to the Administrative UI.
2. Select Infrastructure, X509 Certificate Management, Trusted Certificates and Private Keys.
3. Click Request Certificate.
4. Complete the required fields.
5. Click Save.

A file that conforms to the PKCS #10 specification is generated.

The browser prompts you to save or open the file, which contains the certificate request. If you do not save this file (or open it and extract the text), the Policy Server still generates the private key and self–signed certificate pair. To get a new request file for the private key, generate a new certificate signing request using the Generate CSR feature.

Import a Signed Certificate Response

After completing a certificate request and sending it to the Certificate Authority, the Certificate Authority issues a signed certificate response.

Import the signed certificate into the certificate data store to replace the existing self-signed certificate entry of the same alias.

Follow these steps:

1. Select Infrastructure, X509 Certificate Management, Trusted Certificates and Private Keys.
2. In the list, locate the self-signed certificate that you want to update.
3. Select Action, Update Certificate next to the self-signed entry.
4. Browse to the file you want. You can use a:
   - .p7 or .p7b file that contains the signed certificate and the corresponding certificate chain.
   - .cer or .crt file (base64 PEM file) with the signed certificate without the certificate chain.
5. Select the appropriate entry.
6. Review the certificate information and click Finish.

The signed certificate is imported into the certificate data store and the self-signed certificate is replaced.

Generate a New Certificate Signing Request

A certificate signing request (CSR) is a message that you send to a Certificate Authority to apply for a digital identity certificate. After you create a private key, you can generate a CSR. The CSR contains the public key.

You can generate a new CSR for a self-signed or CA-signed private key/certificate pair. The private key always generates an identical CSR without modifying the existing private key. You generate a new request for an existing private key for the following reasons:

- You no longer have the original request that was generated for the private key/self-signed certificate pair.
- You need a new certificate for an expiring one, which requires a new copy of a CSR to submit to a Certificate Authority.

Follow these steps:

1. Log in to the Administrative UI.
2. Select Infrastructure, X509 Certificate Management, Trusted Certificates and Private Keys.
3. Select Action, Generate CSR for the private key entry for which you want a new CSR.

   A file that conforms to the PKCS #10 specification is generated.
4. Save the CSR when prompted.
5. (Optional) If you require a CA-signed certificate, contact a Certificate Authority. Follow the procedure the Certificate Authority requires for submitting a request. Use the PKCS#10 file you saved in the previous step for the request.

After you complete the certificate request process, the Certificate Authority issues a signed certificate response that you import into the certificate data store. The Policy Server replaces the existing certificate entry of the same alias with the newly imported certificate.

Update Certificates in the Certificate Data Store

You can update key/certificate pairs and standalone certificates in the following ways:

- Update an expiring trusted certificate by deleting the existing certificate and importing a new trusted certificate. The new certificate must match the expiring certificate in the certificate data store.
- Update the certificate by importing a signed trusted certificate or a PKCS7-signed response. The new certificate must match the expiring certificate in the certificate data store.
- Update a certificate with a certificate from a PKCS#12 file. The new private key and certificate pair must match the expiring key/certificate pair in the certificate data store.

The new certificate must be valid before the Policy Server can use it to update an expiring certificate. Certificates are updated and become available immediately after they are imported. If the new certificate is not valid, as determined by its validity interval, the Policy Server cannot use the new certificate.

For importing only a trusted certificate, use a file containing the certificate in a PEM or DER encoding. The standard extension for files of these types is *.crt or *.cer. If the file ends in .p12 or .pfx, it is processed as a certificate data store file containing key/certificate pairs. Finally, if a file ends in .p7 or .p7b, it is processed as a signed response file. Anything else is treated as a certificate file, and CA SiteMinder tries to load a certificate from it.

Note: If you update certificates for a federated environment, you do not have to update any federation objects that use the expiring certificates.

Create the IdP and SP Entities

When CA CloudMinder is providing the identity information for the user, it is acting as the local IdP. The business partner, for example, SalesForce.com, is the remote SP.

Each partner in a federation partnership is considered a *federation entity*. Before you establish a partnership, define a local entity that represents the local partner and a remote entity that represents the remote partner.

The two ways to configure a federation entity are:

- Create an entity without using metadata (see page 26).
- Create an entity by importing metadata (see page 31).

Create an Entity without Using Metadata

Create an entity without metadata by using the following process:

1. Indicate an entity type.
2. Configure the specifics about that entity type.
3. Confirm the entity configuration.

Entity Type Choice

The first step in configuring an entity is to establish the entity type and determine the entity role.

To establish the entity type

1. Log in to the Administrative UI.
2. Select Federation, Partnership Federation, Entities.
3. Click Create Entity.

   The Create Entity dialog displays.

   Note: Click Help for a description of fields, controls, and their respective requirements.
4. Select *one* of the following options:

   Local

   Indicates that you are creating an entity that is local to your site.

   Remote

   Indicates that you are configuring an entity that represents the partner at the remote site.

5. Configure the remaining fields:

   New Entity Type

   Select the asserting or relying party.

   SAMLToken Type (WS-FED only)

   Select the token type, which defines the SAML format for the encrypted token that contains user credential information. Choose the Legacy option only if you want the token to comply with the SAML token type for WS-Federation 1.0.

6. Click Next to configure specifics about the entity.

Detailed Local Entity Configuration

After you have specified the entity type, configure the details of the entity. For a local entity, define the following information:

- Identification information about the entity
- Signature and encryption options
- Name ID formats and attributes

Follow these steps:

1. Begin at the Configure Entity step.
2. Complete any required fields for features and services for the local entity type you are configuring.

   Click Help for a description of the fields.

3. Click Next.

   The Confirm dialog is displayed.

Be aware of the following features:

Entity ID and Entity Name Settings

If the Entity ID represents a remote partner, the value must be unique. If the Entity ID represents a local partner, it can be reused on the same system.

The Entity Name identifies an entity object in the policy store. The Entity Name must be a unique value. This value is for internal use only; the remote partner is not aware of this value.

Note: The Entity Name can be the same value as the Entity ID, but do not share the value with other entities at the same site.

Signing and Encryption Features

For signing and encryption features, you must have the appropriate key/certificate entries in the certificate data store. If you do not have the appropriate key/certificate entries, click Import to import a private key/certificate pair from a file on your local system. You can also import trusted certificates.

Note: If you are using SAML 2.0 POST profile, signing assertions is required.

WSFED Attributes (WS-Federation only)

You can specify various service URLs and IDs for WS-Federation entites to communicate.

Name ID Formats

You can indicate the identifier types that the federated entity supports.

Assertion Attribute Configuration (asserting partners only)

You can configure the asserting party to include specific assertion attributes when it generates an assertion. The recommended method is to define these attributes at the entity level. The entity serves as a template for the partnership so any assertion attributes you define for the entity get propagated to the partnership. The benefit of defining assertion attributes at the entity is that it enables you to use an entity in more than one partnership.

If you want to add or remove assertion attributes for the partnership, make such modifications at the partnership level, not at the entity level.

Detailed Remote Entity Configuration

After you have specified the entity type, configure the details of the entity. For a remote entity type, define the following information:

- Identification information about the entity
- Signature and encryption options
- NameID and attribute information

Follow these steps:

1. Begin at the Configure Entity step.
2. Specify the Assertion Consumer Service URL. Examples:
   - If the SP is a site such as Google, the URL can be similar to:

     https://www.google.com/a/example.com/acs
   - If the SP is a site such as Salesforce.com, the URL can be similar to:

     https://login.salesforce.com/?saml=EK05LGnm40H7
   - If the SP is another business partner, the URL can be similar to:

     http://myserver.forwardinc.com:9080/samlsp/acs

3. Complete any other required fields for features and services for the remote entity type.

   Click Help for the field descriptions.

4. Click Next.

   The Confirm dialog is displayed.

Be aware of the following features:

Entity ID and Entity Name Settings

If the Entity ID represents a remote partner, the value must be unique. If the Entity ID represents a local partner, it can be reused on the same system.

The Entity Name identifies an entity object in the policy store. The Entity Name must be a unique value. This value is for internal use only; the remote partner is not aware of this value.

Note: The Entity Name can be the same value as the Entity ID, but do not share the value with other entities at the same site.

Signing and Encryption Features

For signing and encryption features, you must have the appropriate key/certificate entries in the certificate data store. If you do not have the appropriate key/certificate entries, click Import to import a private key/certificate pair from a file on your local system. You can also import trusted certificates.

Note: If you are using SAML 2.0 POST profile, signing assertions is required.

WSFED Attributes (WS-Federation only)

You can specify various service URLs and IDs for WS-Federation entites to communicate.

Name ID Formats

You can indicate the identifier types that the federated entity supports.

Assertion Attribute Configuration (asserting partners only)

You can configure the asserting party to include specific assertion attributes when it generates an assertion. The recommended method is to define these attributes at the entity level. The entity serves as a template for the partnership so any assertion attributes you define for the entity get propagated to the partnership. The benefit of defining assertion attributes at the entity is that it enables you to use an entity in more than one partnership.

If you want to add or remove assertion attributes for the partnership, make such modifications at the partnership level, not at the entity level.

Confirm the Entity Configuration

Review the entity configuration before saving it.

Follow these steps:

1. Review the settings in the entity dialog.
2. Click Back to modify any settings from this dialog.
3. Click Finish when you are satisfied with the configuration.

A new entity is configured.

Editing Entities from the Partnership

You can click Get Updates next to the local and remote entity fields to update information about the entity. When you select Get Updates, the system asks to pull in the latest information from the entity.

After confirmation, the partnership you are editing is refreshed with the latest entity information. Changes are saved when you complete the partnership wizard. If you do not confirm the update, the partnership configuration remains the same.

The Entity Name identifies an entity object for in the policy store. The Entity Name must be the unique identifier because the product uses this value internally to distinguish an entity. This value is not used externally and the remote partner is not aware of this value.

If the Entity ID represents a remote partner, the value must be unique. If the Entity ID represents a local partner, it can be reused on the same system.

Note: The Entity Name can be the same value as the Entity ID, but do not share the value with any other entity.

An entity is a key component of a federation partnership. Changing an entity alters the partnership significantly; therefore, the Administrative UI does not let you replace an entity after it is in a partnership. To replace an entity, create a partnership.

To provide some flexibility within partnership configuration, you can change an entity ID because it does not identify the entity uniquely. Changing the entity ID at the partnership level does not link the partnership to another entity. The original entity in the partnership does not change. Modifications to an entity are a one-way propagation from the entity to the partnership. A change to the entity ID at the partnership does not get propagated back to the original entity.

Regard entity configurations as templates. Partnerships are created based on the entity templates so changing the partnership does not change the original entity template.

Create an Entity by Importing Metadata

You can import data from a metadata file to create a federation entity. Importing the metadata reduces the amount of configuration for creating a partnership.

You can use metadata in the following ways:

- Import data from a remote partner to create a new remote entity.
- Import data from a remote partner to update an existing remote entity.
- Import data from a local entity to create a new local entity.

  This option can be useful to facilitate a migration from another federation product.

  Note: Federation does not support metadata imports to update or restore an existing partnership and local entity. To update an existing local entity, edit the entity and modify the settings that you want to change. You can import metadata only to create a *new* local entity.

The process for creating a metadata-based entity is as follows:

1. Select a metadata file for configuring a new entity.
2. Select an entity entry from the metadata file. The file can include several entities, but one entity per file is recommended.
3. (Optional) Select the certificates to import into the certificate data store. The certificates must be in the metadata file.

   These certificates can be used for authentication request verification, single logout response verification (SAML 2.0), and encryption (SAML 2.0).
4. Confirm the entity configuration.

Details about these steps are described in the next sections.

Metadata File Selection

The first step to create an entity from metadata is to select the metadata file.

Follow these steps:

1. Log in to the Administrative UI.
2. Select Federation, Partnership Federation, Entities.
3. Click Import Metadata.

   The Import Metadata dialog opens.

   Click Help for the field descriptions.
4. Browse for the metadata file you want to use to create the entity.

5. Select whether to create a new local or remote entity, or update an existing remote entity.

   Note: The Policy Server does not support metadata imports to update an existing partnership and local entity. You can only create a new local entity. To update an existing local entity, edit the entity and modify the settings that you want to change. You can update the existing remote entities or you can create new remote entities.

6. Click Next to select entities from the file.

If you select a metadata file with expired entries, the next dialog that the UI displays contains a section listing the expired entries. You cannot select these expired entries; they are displayed for your reference. If all entities in a metadata file are expired, no entities are displayed. In this case, upload a new document.

Select an Entity to Import

This procedure assumes that you have already selected a metadata file to create an entity. Select the entity from the file.

Follow these steps:

1. Specify a name for the new entity in the Select Entity Defined in File dialog.

   If you are doing a local import to create an entity, define the partnership name.

2. Click on the option button to select the entity.

3. Click Next.

   The Import Certificates dialog displays if importing metadata for a remote entity and the document includes certificate data.

   If the metadata file that you imported contains certificate entries, you can import these entries.

Certificate Imports

To verify signed assertions, import certificates if the metadata includes them. If the metadata does not include certificates, skit this step and go to the Confirm step.

Follow these steps:

1. From the Import Certificates step, select the certificate entry or entries from the metadata file that you want to import.

   If you select a certificate file with invalid entries, the next dialog contains a section listing the expired entries. You cannot select these expired entries. They are displayed for your reference. If all entries in the file are invalid, the import wizard skips the certificate selection step.

   Specify a unique alias for each entry that you chose.

2. Click Next

   The Confirm dialog displays showing a table of entries.

   You can select two entries from a metadata file that have the same certificate. For SAML 1.1 metadata, every entry shows Signing as the usage for the certificate because SAML 1.1 does not encrypt data.

   For SAML 2.0, each entry can show a different usage for the certificate, for example, one for signing, one for encryption. When you get to the Confirm step, the window shows a table with a single certificate entry. The certificate usage is listed as Signing and Encryption. This entry is the combination of the two entries you chose previously. This entry also uses the first alias that you specified for the certificate entry you selected.

   This situation occurs only if the same certificate was listed in the metadata file for both uses. If the file contains two separate certificates, the confirmation step shows both entries in the table.

   For example, you select two entries from the metadata file and you do not realize they are the same certificate. The first usage is Signing and you assign it the alias **cert1.** The second usage is Encryption and you assign it the alias **cert2**. When you confirm the import, you see a table titled Selected Certificate Data with an entry similar to the following entry:

Alias	Issued To	Usage
cert1	Jane Doe	Signing and Encryption

   If no usage is specified in the metadata file, then the usage defaults to Signing and Encryption.

3. Click Next to finish the configuration.

Confirm the Entity Configuration

Review the entity configuration before saving it.

Follow these steps:

1. Review the settings in the entity dialog.
2. Click Back to modify any settings from this dialog.
3. Click Finish when you are satisfied with the configuration.

A new entity is configured.

Establish a User Directory Connection

Partnership federation looks up entries in a user directory to verify identities and retrieve user attributes for a given principal. At the asserting party, the federation partner generates assertions for the appropriate users, and authenticates each user against a user directory. At the relying party, the federation partner extracts the necessary information from an assertion and looks in the user directory for the appropriate user record.

Configure connections to existing user directories by selecting Infrastructure, Directory, User Directories in the Administrative UI. You are only establishing a connection to an existing user directory. You are not configuring a new user directory.

Note: To use an ODBC database in your federated configuration, set up the SQL query scheme and valid SQL queries before selecting an ODBC database as a user directory.

Configure connections to more than one directory if necessary. The directories do not have to be the same type.

For detailed information about user directories, see the *Policy Server Configuration Guide.*

For deployments that use the internal cloud user directory, connect to the internal user directory.

Configure the Local IdP-to-Remote SP Partnership

After you create federation entities, follow the partnership wizard to configure the IdP ->SP partnership. The wizard begins with the basic partnership parameters.

Follow these steps:

1. Select Federation, Partnership Federation, Partnerships.
2. Click Create Partnership.
3. Select SAML2 IdP -> SP.

   Selecting this option indicates that you are the local IdP.

   You come to the first step in the partnership wizard.

4. Complete the following fields

   Partnership Name

   Local IDP ID

   Enter the ID for the local IdP. For this scenario, CA CloudMinder is the Identity Provider. Example: cloudhost.ca.com.

   Remote SP ID

   Enter the remote SP ID. For example, Salesforce.com

   Base URL

   Enter the base URL of the local IDP. For example, http://cloudhost.ca.com:9090

   Skew Time (Seconds)

   Accept the default

5. Move the cloud host directory from the Available Directories list to the Selected Directories list.
6. Click Next to go to the Federation User step.

Configure Assertion Options

Configure assertion options in the Assertion Configuration step of the partnership wizard.

Follow these steps:

1. Configure the settings in the Name ID section.

   The relying party uses these values to know how to interpret the value that is passed in the assertion.

   Based on the value of the NameID Type, complete one of the following tasks:

   - If you selected Static or User Attribute for the Name ID type, complete the Value field.
   - If you selected the DN Attribute for the Name ID type, complete the Value and the DN specification fields.

   Note: Click Help for a description of fields, controls, and their respective requirements.

2. (Optional - SAML 2.0 only) Select Allow Creation of User Identifier so the asserting party can create a value for the NameID. For this feature to work, the AuthnRequest from the relying party must include an AllowCreate attribute.

   Note: If you select this option, the value of the Name ID Format value must be Persistent Identifier.

3. (Optional) Click Add Row in the Assertion Attributes table to specify one or more attributes for inclusion in the assertion. Optionally, you can encrypt the attribute.

   Click Help for detailed information about the columns in the attribute table.

   Note: For attributes from an LDAP user store, you can add multivalued user attributes to an assertion. The Help describes how to specify multivalued user attributes.

4. (Optional) If you have written an assertion generator plug-in using the CA SiteMinder® Federation Java SDK, complete the fields in the Assertion Generator Plug-in section.

   To write a plug-in, see the *Programming Guide for Federation Manager Java SDK*.

5. Click Next to continue with partnership configuration.

Single Sign-on Configuration (Asserting Party)

Configure single sign-on at the asserting party to specify how the asserting party delivers an assertion to a relying party.

Follow these steps:

1. Begin at the appropriate step in the partnership wizard.

   SAML 1.1 and WS-FED

   Single Sign-On

   SAML 2.0

   SSO and SLO

   Any values that are defined during the creation or import of the remote relying party are filled in.

   Note: Click Help for a description of fields, controls, and their respective requirements.

2. In the Authentication section, configure the following fields so CA CloudMinder can act as the IdP

   Authentication Mode

   Delegated

   Delegated Authentication Type

   Cloud

   Delegated Authentication URL

   Enter the URL of the system authenticating the user requesting a resource. Use the following syntax for the delegated URL:

   http://*cloud_system*:*port*/chs/login/*tenant_name*/*application_name*

   The *cloud_system* is the system where the user console is installed.

   Example URL:

   http://cmserver.fowardinc.com:832/chs/login/tenant1/confidential_app

   Configure AuthnContext

   Use Predefined Authentication Class

   Authentication Class field

   Supply a static URI for SAML 1.1, SAML 2.0, and WS-FED.

   Additionally, for SAML 2.0 only, the system can automatically detect an authentication class. The URI is placed in the AuthnContextClassRef element in the assertion to describe how a user is authenticated.

3. Complete the fields in the SSO section to determine how single sign-on operates. These settings let you control the following features:

   - Single sign-on binding
   - Assertion validity

     The SSO Validity Duration and the Skew Time determine when the assertion is valid. Read the information about assertion validity (see page 104) to understand how these settings work together.

   For SAML 2.0, you can configure these features:

   - Initiation of single sign-on from which partner
   - SP session validity
   - SP session duration
   - User consent to share identity information with the SP

   Note: Click Help for a description of fields, controls, and their respective requirements.

4. Specify the URL for the Remote Assertion Consumer Service. This service is the service at the relying party that processes received assertions.

   Your partner needs to supply this URL to you.

5. If you selected HTTP-Artifact, configure the back channel settings (see page 39).

Assertion Validity for Single Sign-on

For single sign-on, the values of the Skew Time and the SSO Validity Duration determine how long an assertion is valid. The Policy Server applies the skew time to the generation and consumption of assertions. In the assertion document, the NotBefore and NotOnOrAfter values represent the beginning and end of the validity interval.

At the asserting party, the Policy Server sets the assertion validity. The Policy Server determines the beginning of the validity interval by taking the system time when the assertion is generated. The software sets the IssueInstant value in the assertion from this time. The Policy Server then subtracts the skew time value from the IssueInstant value. The resulting time becomes the NotBefore value.

NotBefore=IssueInstant - Skew Time

To determine the end of the validity interval, the Policy Server adds the Validity Duration value and the skew time to the IssueInstant value. The resulting time becomes the NotOnOrAfter value.

NotOnOrAfter=Validity Duration + Skew Time + IssueInstant

Times are relative to GMT.

For example, an assertion is generated at the asserting party at 1:00 GMT. The skew time is 30 seconds and the validity duration is 60 seconds, making the assertion validity interval between 12:59:30 GMT and 1:01:30 GMT. This interval begins 30 seconds before the time the assertion was generated and ends 90 seconds afterward.

At the relying party, the Policy Server performs the same calculations as it does at the asserting party to determine if the assertion it receives is valid.

Calculating Assertion Validity when CA SiteMinder is at Both Sides of the Partnership

The total time the assertion is valid is the sum of the SSO validity duration plus two times the skew time. The equation is:

Assertion Validity = 2x Skew Time (asserting party) + SSO Validity Duration + 2x Skew Time (relying party)

The initial part of the equation (2 x Skew Time + SSO Validity Duration) represents the validity window at the asserting party. The second part of the equation (2 x Skew Time) represents the skew time of the system clock at the relying party. You multiply by 2 because you are accounting for the NotBefore and the NotOnOrAfter ends of the validity window.

Note: For the Policy Server, the SSO Validity Duration is only set at the asserting party.

Example

Asserting Party

The values at the asserting party are as follows:

IssueInstant=5:00PM

SSO Validity Duration=60 seconds

Skew Time = 60 seconds

NotBefore = 4:59PM

NotOnOrAfter=5:02PM

Relying Party

The relying party takes the NotBefore and NotOnOrAfter values that is receives in the assertion then applies its skew time to calculate new values.

Skew Time = 180 seconds (3 minutes)

NotBefore = 4:56PM

NotOnOrAfter=5:05PM

Based on these values, the calculation for the total assertion validity window is:

120 seconds (2x60) + 60 seconds + 360 seconds (2x180) = 540 seconds (9 minutes).

Back Channel Authentication for Artifact SSO

Artifact single sign-on requires the relying party to send an artifact to the asserting party to retrieve the assertion. The asserting party uses the artifact to retrieve the correct assertion and returns the assertion to the relying party over a back channel.

You can require an entity to authenticate to access the back channel. The back channel can also be secured using SSL, though SSL is not required.

Securing the back channel using SSL involves:

1. Enabling SSL.

   SSL is not required for Basic authentication but you can use Basic over SSL. SSL is required for Client Cert authentication.

2. Configuring an incoming or outgoing back channel for the SAML 2.0 communication exchange. The direction you configure depends on the role of the local entity.

   Configuring separate channels is supported only for SAML 2.0. The back channel configuration for SAML 1.1 artifact single sign-on uses a single configuration for each partnership. CA SiteMinder uses the correct direction automatically (incoming for a local producer and outgoing for a local consumer).

   Select which direction to configure for SAML 2.0 single sign-on based on the entity you are configuring.

   - The local asserting party uses the incoming channel.
   - The local relying party uses the outgoing channel.

   Note: You can configure an incoming and outgoing back channel; however, a channel can have only one configuration. If two services use the same channel, these two services use the same back channel configuration. For example, if the incoming channel for a local asserting party supports HTTP-Artifact SSO and SLO over SOAP, these two services must use the same back channel configuration.

3. Choosing the type of authentication for the relying party to gain access across the protected back channel. The authentication method applies per channel (incoming or outgoing).

   The options for back channel authentication are:

   - Basic
   - Client Cert
   - NoAuth

   The Administrative UI help describes these options in detail.

   Important! The authentication method for the incoming back channel must match the authentication method for the outgoing back channel on the other side of the partnership. Agreeing on the choice of authentication method is handled in an out of band communication.

Configure the HTTP-Artifact Back Channel

Protect the HTTP-artifact back channel across which the asserting party sends the assertion to the relying party.

Consider the following limitation:

You cannot use client certificate authentication with the following web servers running ServletExec:

- IIS web servers at a CA SiteMinder producer/Identity Provider because of a limitation in IIS.
- SunOne/Sun Java Server web servers at a CA SiteMinder producer/Identity Provider because of a documented limitation in ServletExec.

Follow these steps:

1. Begin at the Back Channel section in the Single Sign-on or the SSO and SLO step of the partnership wizard.
2. Select HTTP-Artifact in the SSO section.

   The Authentication Method field becomes active.
3. Select the type of authentication method for the incoming or outgoing back channel, or both.

   Click Help for the field descriptions.

   - If you select the client certificate authentication scheme, add a private key/certificate pair to the certificate data store. The private key/certificate pair is issued from a Certificate Authority.

     Important! The CN of the Subject in the certificate must be the same as the partnership name in the producer to consumer partnership that is configured at the producer.

     For instructions on adding a certificate, see the Policy Server Configuration Guide. Skip this step if the key/certificate pair is already in the data store.
   - If you select No Auth as the authentication method, no additional steps are required.
4. Depending on the authentication method you select, several additional fields are displayed for you to configure.

After entering values for all the necessary fields, the back channel configuration is complete. You can enable SSL on each side of the connection for added security.

Sign and Encrypt Federation Messages

Securing an assertion and encrypting data within the assertion is a critical part of partnership configuration. The Signature step (SAML 1.1) and the Signature and Encryption step (SAML 2.0) let you configure signing and encryption of assertions.

For SAML 2.0, you have the option of choosing a signing algorithm for signing tasks. The ability to select an algorithm supports the following use cases:

- An IdP-->SP partnership in which the IdP signs assertions, responses and SLO-SOAP messages with the RSAwithSHA1, or the RSAwithSHA256 algorithm.
- An SP-->IdP partnership in which the SP signs authentication requests and SLO-SOAP messages with the RSAwithSHA1, or the RSAwithSHA256 algorithm.

Signature verification automatically detects which algorithm is in use on a signed document then verifies it. No configuration for signature verification is required.

Signature Configuration at a SAML 2.0 IdP

The Signature and Encryption step in the partnership wizard lets you define how the product uses private keys and certificates for the following signing functions:

- Sign and verify SAML assertions, assertion responses, and authentication requests.

  For SAML 2.0 POST binding, you are required to sign assertions.
- Sign single logout responses and requests (HTTP-Redirect and SOAP bindings).

There can be multiple private keys and certificates in the certificate data store. If you have multiple federated partners, you can use a different key pair for each partner.

Note: If the system is operating in FIPS_COMPAT or FIPS_MIGRATE mode, all certificate and key entries are available from the pull-down list. If the system is operating in FIPS-Only mode, only FIPS-approved certificate and key entries are available.

To configure signing options

1. Select the Signature and Encryption step in the partnership wizard.
2. In the Signature section, select an alias for the Signing Private Key Alias field. If there is no private key available, click Import to import one. Or, click Generate to create a certificate request.

   By completing this field, you are indicating which private key the asserting party uses to sign assertions, single logout requests and responses.

   Note: click on Help for a description of the fields.

3. Select the hash algorithm for digital signing in the Signing Algorithm field. The IdP signs assertions, responses and SLO-SOAP messages with the specified algorithm.

   Select the algorithm that best suits your application.

   RSAwithSHA256 is more secure than RSAwithSHA1 due to the greater number of bits used in the resulting cryptographic hash value.

   The system uses the algorithm that you select for all signing functions.

4. Select an alias from the certificate data store or the Verification Certificate Alias field.

   By completing this field, you are indicating which certificate verifies signed authentication requests or single logout requests or responses. If there is no certificate in the database, click Import to import one.

5. (Optional) Specify Artifact and POST signature options for the assertion or response or both.

6. (Optional) Specify an SLO SOAP signature option for the logout request, the logout response or both when you are using single logout.

7. (Optional) Select the check box for Require Signed Authentication Requests. This check box verifies that the asserting party only accepts signed requests from the relying party.

   Activate a partnership for all configuration changes to take effect and for the partnership to become available for use. Restarting the services is not sufficient.

If you are using the product in a test environment, you can disable signature processing to simplify testing. Click the Disable Signature Processing check box.

Important! Enable signature processing in a SAML 2.0 production environment.

Encryption Configuration at a SAML 2.0 IdP

The Signature and Encryption step in the Partnership wizard lets you define how the Policy Server uses private keys and certificates to do the following tasks:

- Sign and verify SAML assertions, assertion responses, and authentication requests.

  For SAML 2.0 POST binding, you are required to sign assertions.

- Sign single logout responses and requests (HTTP-Redirect and SOAP bindings).
- Encrypt and decrypt entire assertions, Name IDs and attributes.

There can be multiple private keys and certificates in the certificate data store. If you have multiple federated partners, you can use a different key pair for each partner.

To configure encryption options

1. In the Encryption section, select one or both of the following check boxes to specify the assertion data to be encrypted:
   - Encrypt Name ID
   - Encrypt Assertion
2. Select the certificate alias from the certificate data store for the Encryption Certificate Alias.

   This certificate encrypts assertion data. If no certificate is available, click Import to import one.
3. Select values for the Encryption Block Algorithm and Encryption Key Algorithm fields.

   For the following block/key algorithm combinations, the minimum key size that is required for the certificate is 1024 bits.
   - Encryption Block Algorithm: 3DES

     Encryption Key Algorithm: RSA-OEAP
   - Encryption Block Algorithm: AES-256

     Encryption Key Algorithm: RSA-OEAP

     Note: To use the AES-256 bit encryption block algorithm, install Java Cryptography Extension (JCE) Unlimited Strength Jurisdiction Policy Files. You can download these files from http://java.sun.com/javase/downloads/index.jsp.

The encryption configuration is complete.

Partnership Confirmation

Review the partnership configuration before saving it.

Follow these steps:

1. Review the settings in the Confirm step of the Partnership wizard.
2. Click Modify in each group box to change any settings.
3. Click Finish when you are satisfied with the configuration.

The partnership configuration is complete.

Activate the Partnership

After you configure all the required settings for a partnership, activate it to use it. You can also deactivate a partnership using the same process.

Follow these steps:

1. Select Federation, Partnership Federation, Partnerships.

   The Partnerships dialog opens.

2. From the Actions menu, select Activate or Deactivate next to the partnership of interest.

   A confirm dialog displays.

   Note: Activate is only available for a partnership in DEFINED or INACTIVE status. Deactivate is only available for a partnership in ACTIVE status.

3. Click Yes to confirm your selection.

   The status of the partnership is set and the display is refreshed.

Important! Deactivate a partnership before you modify it.

Proceed with the Authentication Scheme Setup

After you configure the federated partnerships, the final task at the Administrative UI is to configure and apply an authentication scheme to the redirect JSP resource. This resource invokes the necessary authentication scheme and redirects the user to the target application.

Note: Authentication schemes are only required if the application is protected by OpenID, OAuth, or one of the advanced authentication schemes.

After you apply the authentication scheme, the remaining SSO tasks are done from the User Console, including:

- Configuring an authentication method.
- Configuring an application.

Configure and Apply an Authentication Scheme

The authentication schemes that correspond to the advanced authentication flows are preconfigured in the Administrative UI. These authentication schemes are:

- For ArcotID OTP Only: ARCOTOTP_MOBILE_ONLY_AUTH_SCHEME
- For ArcotID OTP with Risk: RISK_AND_ARCOTOTPMOB_ AUTH_SCHEME
- For ArcotID PKI Only: ARCOTID_ONLY_AUTH_SCHEME
- For ArcotID PKI with Risk: RISK_AND_ARCOTID_ AUTH_SCHEME

You establish a one-to-one correspondence between an authentication method that is configured in the User Console and an authentication scheme in the Administrative UI. The authentication method and authentication scheme work together to protect access to the specified application.

The authentication scheme protects the authentication URL that is specified for a given authentication method. To apply the authentication scheme, assign the authentication scheme to a realm and then include the realm in a policy.

Follow these steps:

1. Configure a realm and a rule for the resource (see page 46).
2. Add rules to the tenant policy (see page 48).

Configure a Realm and a Rule for the Resource

A realm groups resources that have similar security requirements and share a common authentication scheme. In the tenant domain, create a realm for each authentication scheme that the tenant administrator wants to use.

Note: The following procedure assumes that you are creating an object. You can also copy the properties of an existing object to create an object.

Follow these steps:

1. Log in to the Administrative UI.
2. Select Policies, Domain, Realms.

   The Realms screen opens.
3. Click Create Realm.
4. Select the tenant domain that you want to modify, and then click Next.

   Note: The tenant domain name is in the *tenant-tag*Domain format.

5. Type a name and description for the realm.

   Specify a name that indicates that the realm is for an authentication URL.

6. Click Lookup Agent/Agent Group.

7. Select **cam-agent** from the list of agents, and then click OK.

8. Specify the resource filter for the authentication scheme. This scheme must tie in to the authentication method chosen in the User Console.

   ArcotID OTP

   For environments created in CA CloudMinder 1.51 or later:

   /chs/redirect/*tenant_tag*/arcototp

   For environments created before CA CloudMinder 1.51:

   /affwebservices/<tenant-name>/arcototp.jsp

   ArcotID OTP with Risk

   For environments created in CA CloudMinder 1.51 or later:

   /chs/redirect/*tenant_tag*/arcototprisk

   For environments created before CA CloudMinder 1.51:

   /affwebservices/<tenant-name>/arcototprisk.jsp

   ArcotID PKI

   For environments created in CA CloudMinder 1.51 or later:

   /chs/redirect/*tenant_tag*/arcotid

   For environments created before CA CloudMinder 1.51:

   /affwebservices/<tenant-name>/arcotid.jsp

   ArcotID PKI with Risk

   For environments created in CA CloudMinder 1.51 or later:

   /chs/redirect/*tenant_tag*/arcotidrisk

   For environments created before CA CloudMinder 1.51:

   /affwebservices/<tenant-name>/arcotidrisk.jsp

   tenant_tag is a unique identifier for a tenant. You specify the tag when deploying a tenant environment in the Administrative UI. To view a list of tags, select the Tenants tab.

9. Complete the remaining fields:

   Default Resource Protection

   Protected

   Authentication Scheme

   Select the authentication scheme that corresponds to the resource filter.

10. Create a rule as follows:
    a. Click Create in the Rules area.

       The Create Rule screen opens.

    b. Enter a name and description for the rule.
    c. Enter the asterisk (*) in the Resource field.
    d. Select Get and Post from the Action list.
    e. Accept the defaults for the remaining settings, and then click OK.

       The rule is created.

11. Specify the session properties.

    Note: Click Help for information about these properties.

12. Skip the other configuration options.
13. Click Finish.

    The realm is configured.

Add Rules to the Policy

Rules indicate which resources are part of a policy and whether to allow or deny access to the resources when the rule fires.

Note: Add at least one rule or rule group to a policy.

Follow these steps:

1. Select Policies, Domain, Domains.

   The Domains screen opens.

2. Click the pencil icon for the tenant domain.
3. Click the Policies tab.
4. Click the pencil icon for the *tenant_tag*_chsauthmethods_policy_es policy.

5. Click the Rules tab.
6. Perform the following steps for each rule that you want to add:
   a. Click Add Rule.

      The Available Rules pane opens.

   b. Select the rule that you created for the authentication URL resource, and then click OK.

      The rule is added to the tenant policy.

Create Authentication Methods

An authentication method represents how an application is protected. After you configure an authentication method, you assign it to the application you want to protect. Multiple applications can use the same authentication method. A single application can reference multiple authentication methods.

Configure an authentication method that satisfies the protection requirements for an application.

Note: The system creates authentication methods corresponding to each of the advanced authentication flows. If you are configuring Advanced Authentication for the tenant, do not create an authentication method. Modify the existing authentication method as described in this procedure.

Follow these steps:

1. Log in to the User Console.
2. Navigate to Applications, Authentication Methods, Create an Authentication method.
3. In the top section of the Create Authentication method screen, complete the following fields:

   Name

   Enter a string that identifies the authentication method you are configuring.

   Description

   Enter a description for the authentication method. The login page displays this description as a label.

   Enabled

   Select this check box to make the authentication method immediately available.

4. In the Configure Authentication Method section, select one of the following options and enter the authentication URL for that option.

   When the authentication method is associated with an application, the authentication service appends the redirect URL for the application.

   Note the following variables in the URLs:

   cloud_host is the CA CloudMinder system.

   local_entity_ID is the name of the local entity that is specified in the IdP-to-SP partnership, which is configured at the CSP console.

   remote_entity_ID, consumer_entity_ID or resource_partner_ID is the name of the remote entity that is specified in the configuration of the asserting-to-relying party partnership. The partnership is configured at the CSP console.

   Basic

   Represents a form-based authentication scheme that uses the basic credentials of a user name and a password. The basic authentication method corresponds to the HTML Forms authentication scheme in the Administrative UI.

   Enter the authentication URL of the following format:

   http://*cloud_host*:*port*/chs/redirectservlet/*tenant_tag*/forms

   tenant_tag is a unique identifier for a tenant. You specify the tag when deploying a tenant environment in the Administrative UI. To view a list of tags, select the Tenants tab.

   External IDP—Google or Facebook

   Represents a third-party identity provider (IdP) that authenticates users. Social media sites, such as Google or Facebook can serve as external IdPs. Other federated partners that support the SAML and WS-Federation protocols can also serve as external IdPs.

   If Google or Facebook is acting as the third-party IdP, specify the OpenID or OAuth authentication method. Each site supports both protocols.

   Enter the relevant URL for the protocol, as shown:

   OpenID

   http://*cloud_host*:*port*/affwebservices/*tenant_tag*/*duplicate_openid_file*.jsp

   When configuring the OpenID authentication scheme at the Administrative UI, the default openid.jsp file is copied and given a unique name, such as openid-google.jsp. Having a unique jsp file is necessary to distinguish OpenID configurations.

   The default JSP file is located in the directory /opt/CA/secure-proxy/Tomcat/webapps/affwebservices/redirectjsp.

OAuth

http://*cloud_host*:*port/*affwebservices/*tenant_tag*/*duplicate_oauth_file*.jsp

When configuring the OAuth authentication scheme in the Administrative UI, the default oauth.jsp file is copied and given a unique name, such as oauth-google.jsp. Having a unique jsp file is necessary to distinguish OAuth configurations.

The default JSP file is located in the directory /opt/CA/secure-proxy/Tomcat/webapps/affwebservices/redirectjsp.

tenant_tag is a unique identifier for a tenant. You specify the tag when deploying a tenant environment in the Administrative UI. To view a list of tags, select the Tenants tab.

External IDP—Other

Select Other when a SAML or WS-Federation-compliant partner is the IdP. The federation profiles SAML 1.1, SAML 2.0, and WS-Federation 1.2 are all supported.

Enter the relevant URL for the protocol, as shown.

For SAML 1.1 transactions
http://*cloud_host.domain:port*/affwebservices/public/intersitetransfer?CONSUMERID=*consumer_entity_ID*&TARGET=http://*consumer_site/target_url*

For SAML 2.0 SP-initiated transactions

http://*cloud_host.domain:port/*affwebservices/public/saml2authnrequest?ProviderID=*local_entity_ID*&RelayState=http://*sp_site/target_url*

For SAML 2.0 IdP-initiated transactions

http://*cloud_host.domain:port/*affwebservices/public/saml2authnrequest?SPID=*remote_entity_ID*&RelayState=http://sp_site/*target_url*

For WS-Federation IP-initiated transaction

http://*cloud_host.domain:port*/affwebservices/public/wsfeddispatcher?wa=wsignin1.0&wtrealm=*resource_partner_ID*&wctx=*target_url*

Advanced Authentication

Represents one of the authentication protocols that the CA CloudMinder Advanced Authentication Service provides.

Select one of the following options and the URL is entered automatically:

For ArcotID PKI Only

For environments created in CA CloudMinder 1.51 or later:

https://cl*oud_host*:p*ort*/chs/redirectservlet/*tenant_tag*/arcotid

For environments created before CA CloudMinder 1.51:

https://cl*oud_host*:p*ort*/affwebservices/<tenant-name>/arcotid.jsp

For ArcotID PKI with Risk

For environments created in CA CloudMinder 1.51 or later:

https://cl*oud_host*:p*ort*/chs/redirectservlet/*tenant_tag*/arcotidrisk

For environments created before CA CloudMinder 1.51:

https://cl*oud_host*:p*ort*/affwebservices/<tenant-name>/arcotidrisk.jsp

For ArcotID OTP Only

For environments created in CA CloudMinder 1.51 or later:

https://cl*oud_host*:p*ort*/chs/redirectservlet/*tenant_tag*/arcototp

For environments created before CA CloudMinder 1.51:

https://cl*oud_host*:p*ort*/affwebservices/<tenant-name>/arcototp.jsp

For ArcotID OTP with Risk

For environments created in CA CloudMinder 1.51 or later:

https://cl*oud_host*:p*ort*/chs/redirectservlet/*tenant_tag*/arcototprisk

For environments created before CA CloudMinder 1.51:

https://cl*oud_host*:p*ort*/affwebservices/<tenant-name>/arcototprisk.jsp

tenant_tag is a unique identifier for a tenant. You specify the tag when deploying a tenant environment in the Administrative UI. To view a list of tags, select the Tenants tab.

5. Click Submit.

The authentication method is available to protect an application.

Creating Roles to Assign Accounts

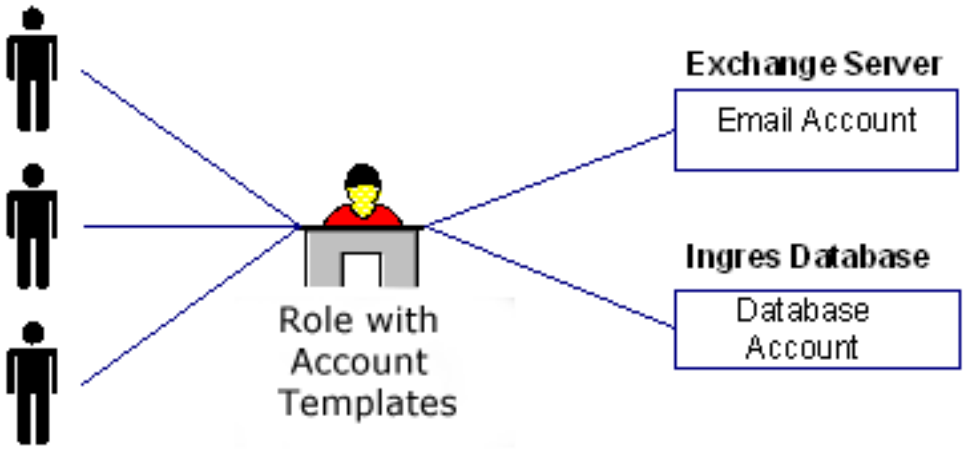
As an administrator, you can configure CA CloudMinder to automatically create a user account in a software resource. For example, you can configure the system to automatically create user accounts in Salesforce.com for employees who need access to this resource. Such a resource is named an *endpoint*.

Note: The following instructions assume that an administrator has already created and configured the endpoint in the system. For more information, see Integrating Managed Endpoints.

In most organizations, administrators spend significant time providing users with login accounts for different systems and applications. To simplify this repetitive activity, you can create provisioning roles, which are roles that contain account templates. The templates define the attributes that exist in one type of account. For example, an account template for an Exchange account defines attributes such as the size of the mailbox. Account templates also define how user attributes are mapped to accounts.

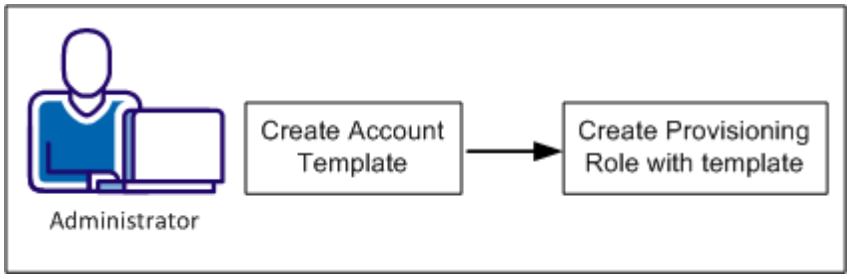
Consider an example where every employee at Forward, Inc needs access to a database and email. An administrator wants to avoid creating a database account and an email account for each employee one at a time. Therefore, the administrator creates a provisioning role for that company. The role contains an account template for a Microsoft Exchange server, to provide email accounts, and a template for an Oracle database. In this example, the Exchange server and the Oracle database are named endpoints, which are the system or application where the accounts exist.

Note: Forward, Inc. is a fictitious company name which is used strictly for instructional purposes only and is not meant to reference an existing company.



After the roles are created, business administrators, such as managers or support personnel, can assign those roles to users to give them accounts in endpoints. After users receive the role, they can log in to the endpoint.

Creating a provisioning role that includes an account template is a two-step process as follows:



The following sections explain how to create a role that can be used to assign accounts:

1. Create an Account Template (see page 54)
2. Create a Provisioning Role (see page 57)

Create an Account Template

A default account template exists for each endpoint type. In a provisioning role, you can use the default account template. However, you can create your own account templates for any endpoint that you have configured.

Follow these steps:

1. Log in to the User Console and select Endpoints, Manage Account Templates, Create Account Template.

   A screen appears with a list of endpoint types.

2. Select an endpoint type for the template.
3. Complete the Account Template tab.
   a. Provide an account template name.
   b. Select Use Strong Synchronization for the maximum correlation of the account template and endpoint account.
4. Complete the Endpoints tab.
   a. Select an endpoint.
   b. Define Endpoint Name as the system name of the endpoint or localhost if that applies.
5. Complete the Account tab.
   a. Modify the rule strings (see page 55) in percent signs if necessary. The rules strings define the format of Login fields for the account.
   b. Enter a %AC% rule string in the Account Name field. You enter this string because account names must be unique.

6. Complete the fields in the other tabs or use the default values.

   Each endpoint type has a different set of tabs. Click Help for field definitions.

7. Click Submit.

CA CloudMinder creates the account template and makes it available for use in provisioning roles.

Rule Strings in Account Templates

When you create an account template, you use rules strings to define the format of many account attributes. Rule strings are variables for the actual value. Rules strings are useful when you want to generate attributes that change from one account to another. When rules are evaluated, Identity Management replaces the rule strings entered in the account templates with data specified in the user object.

Note: Rule evaluation is not performed on accounts created during an exploration or on accounts created without provisioning roles.

The following table lists the rule strings in Identity Management:

Rule String	Description
%AC%	Account name
%D%	Current date in the format *dd/mm/yyyy* (the date is a computed value that does not involve the global user information*).* This rule string is equivalent to one of the following: %$$DATE()% %$$DATE%
%EXCHAB%	Mailbox hide from exchange address book
%EXCHS%	Mailbox home server name
%EXCMS%	Mailbox store name
%GENUID%	Numeric UNIX/POSIX user identifier. This rule variable is the same as %UID% as long as the global user UID value is set. However, if the global user has no assigned UID value, and UID-generation is enabled (Global Properties on System Task), several actions occur. The next available UID value is allocated, assigned to the global user, and used as the value of this rule variable.
%P%	Password

Rule String	Description
%U%	Global user name
%UA%	Full address (generated from street, city, state, and postal code)
%UB%	Building
%UC%	City
%UCOMP%	Company name
%UCOUNTRY%	Country
%UCU*xx*% or %UCU*xxx*%	Custom field (*xx* or *xxx* represents the two-digit or three-digit field ID as specified on the Custom User Fields tab in the System Task frame)
%UD%	Description
%UDEPT%	Department
%UE%	Email address
%UEP%	Primary email address
%UES%	Secondary email addresses
%UF%	First name
%UFAX%	Facsimile number
%UHP%	Home page
%UI%	Initials
%UID%	Numeric UNIX/POSIX User Identifier
%UL%	Last name
%ULOC%	Location
%UMI%	Middle initial
%UMN%	Middle name
%UMP%	Mobile telephone number
%UN%	Full name
%UO%	Office name
%UP%	Telephone number
%UPAGE%	Pager number
%UPC%	Postal code, ZIP Code
%UPE%	Telephone number extension

Rule String	Description
%US%	State
%USA%	Street address
%UT%	Job title
%XD%	Generates the current timestamp in XML dateTimeValue format, a fixed-length string format. In a dateValue or timeValue attribute, you can write an (:offset,length) substring expression to extract the date or time parts of the dateTimeValue. For example, %XD:1,10% yields YYYY-MM-DD; and %XD:12,8% yields HH:MM:SS.

Create a Provisioning Role

After you create the account template, you decide about the role requirements, as follows:

- The accounts that apply to the role
- Who can assign this role
- Who can modify this role

After you decide about the role requirements, you are ready to create a provisioning role.

Follow these steps:

1. Log in to the User Console and click Roles and Tasks, Provisioning Roles, Create Provisioning Role.
2. Complete the Profile tab.

   Only the Name field is required unless you are using a customized version of Create Provisioning Role.
3. Complete the Account Templates tab.
   a. Click an endpoint type, such as SAP.
   b. Click an account template.

      The templates that you can click are based on the endpoint type you selected.
   c. Add more account templates if needed for different endpoint types.
4. Complete the Administrators tab and Owners tab.

   Add admin rules that control who manages members and administrators of this role.

   Add owner rules that control who can modify this role.

5. Click Submit.

   A message appears to indicate the status of the Create Provisioning Role task.

6. To verify that the role was created, click Roles and Tasks, Provisioning Roles, View Provisioning Role.

You have now successfully created a provisioning role. The role can now be assigned to users, so that they can access the accounts that they need.

Create an Application

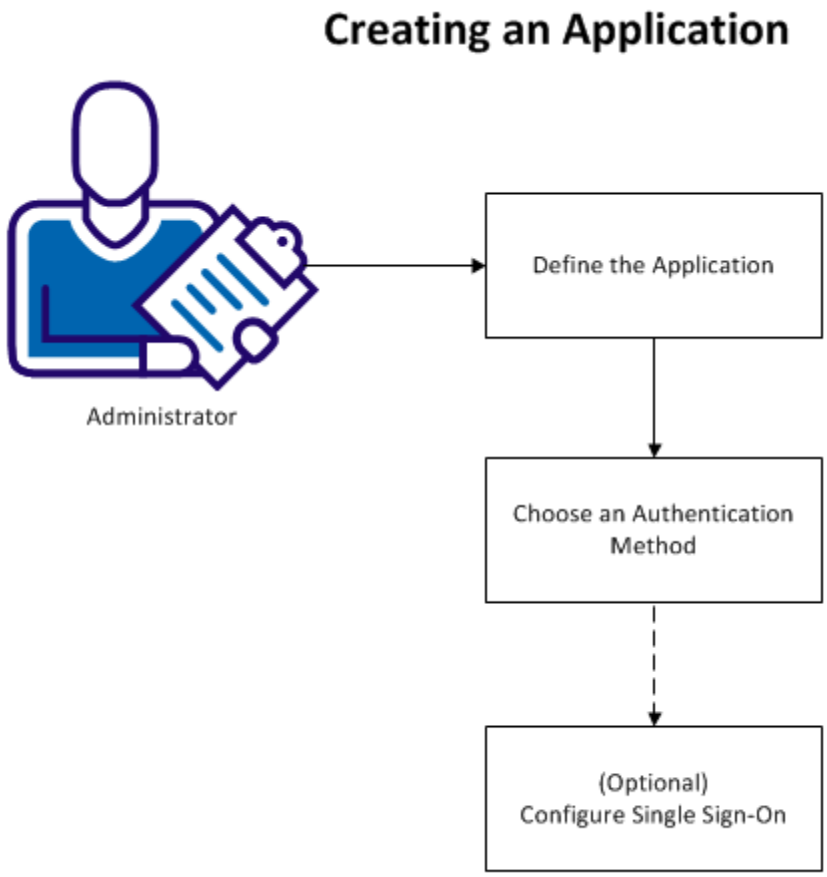
As an administrator, you want to give your users secure and convenient access to software resources. For example, your users need access to your email system, which can be hosted on-premise by your organization. Users also need access to Salesforce.com, which an external organization hosts in the cloud.

Note: Although a tenant administrator typically creates applications for their environment, a hosting administrator can also perform this task.

You create an application to define how users access a software resource. For example, when you configure an application, you define what type and level of security protects the resource. If you have purchased CA CloudMinder Advanced Authentication, you can configure advanced security such as two-factor authentication to protect the resource. If you have purchased the CA CloudMinder Single Sign-on service, you can configure SSO for the application. Users only log in once to access all applications that are configured for SSO.

Once an application is configured, you can give users access to the software resource. You can configure a service that includes the application, and can assign the service to users. The users can click the icon in the User Console Home Page to access the application. For more information, see Creating a Service Using the Service Wizard (see page 64). You can also give users access to the application by making a link to the application available. For example, you can insert the link into any web page or you can send the link in an email.

The following diagram shows the information to understand and the steps to perform to create an application and make it available to users.



The following topics explain how to create an application:

1. Define the Application Profile (see page 59)
2. Choose an Authentication Method (see page 61)
3. (Optional) Configure Single Sign-On (see page 63)

Define the Application

You define the application details through the User Console.

Follow these steps:

1. Log in with an account that has application management privileges.

   For example, the default Tenant Administrator role has the appropriate privileges.

2. From the navigation menu, select Applications.
3. Click Applications, then Create Application.

   The Create Application screen appears.
4. Enter a name and description.
5. Associate a group with the application, if desired.

   Only the users who are members of the indicated group receive access.

   Note: If you are configuring the application for SSO access, the group that you choose must match the group name that is indicated in the SSO partnership configuration for this application. Only if the group names match will the system restrict access to group members. To confirm the group name that is indicated in the partnership configuration, refer to your hosting administrator. SSO partnership configuration information is available in the CSP Console.
6. Enter a launch URL for the Application.

   A launch URL is the fully qualified domain name of the software resource you want to make available to users. For example, if a user clicks the icon for this application in the User Console Home page, they are directed to the launch URL.

   If you are configuring the application for SSO access, the launch URL is the SSO Service URL generated during SSO partnership configuration. Refer to your hosting administrator for this information.

   If you are not configuring an SSO application, simply enter the fully qualified domain name of the software resource. Use the following format:

   https://softwareresourcedomainname.com
7. Choose a logo.

   This logo is the icon for the application that appears in the User Console Home page. Users can click the icon to access the software resource.

   Note: You can also give users access to the application by inserting a link to the application into any web page.
8. Enter a welcome message.

   When users click any link you provide to the application, a login screen appears. The welcome message appears at the top of this screen.

9. Select a self-registration task.

   If a user attempts to access the application but the user does not have a CA CloudMinder account, you can allow them to self-register. Choose one of the following self-registration tasks:

   Create New Account

   Presents a simple registration form. Upon submission, creates a user account.

   Create New Account with Workflow

   Presents a simple registration form. Upon submission, forwards the user account request to one or more approvers. Creates an account upon approval.

   Create New Account with Domain Validation

   Presents a simple registration form. Upon submission, compares the email domain of the user to the tenant email domain. If they match, sends a confirmation email to the user. Creates an account upon user confirmation.

   Note: The tenant email domain is specified in the User Console, under Tenant Administration, Tenant Settings.

   Self-Registration with Attribute Exchange

   Do not choose this self-registration task in the context of application access. This task is intended for a separate purpose.

10. Choose an authentication method. (see page 61)

Choose an Authentication Method

In the Create Application screen, continue the process of creating an application by choosing one or more authentication methods. When a user attempts to access the application, the system presents a login screen. The authentication methods that you choose appear on this screen. The user can log in using their choice of the available authentication methods.

For example, you can select the Basic and Google External IDP authentication methods for an application. The application login screen displays user name and password fields for basic authentication. The login screen also displays the Google icon, so users can log in with their Google credentials.

Follow these steps:

1. In the Authentication Methods area, click Add.

   The Select Authentication Methods screen displays a list of the authentication methods available in the tenant environment.

   Note: First, create authentication methods in the system before you perform this step. You define authentication methods through the User Console, using the Authentication Methods tasks. For more information, see Create Authentication Methods.

2. Select one or more authentication methods. The following types of authentication method are available:

   Basic

   Offers simple user name and password login.

   External IDP

   Offers log in through an external credential provider, such as Google or Facebook.

   Advanced Authentication

   Offers advanced authentication methods that have been configured for your environment, such as One Time Password (OTP) authentication.

   Note: Advanced Authentication methods only appear if you have purchased the Advanced Authentication Service.

   You can choose as many authentication methods, of any type, as you want. All the methods that you select are displayed on the login page that appears when a user attempts to access the application.

3. Click Select.

   The Create Application screen appears, updated with the list of authentication methods you selected.

4. (Optional) From the drop-down list, choose a default authentication method.

   Note: Advanced Authentication methods are never available as a default.

5. Configure Single Sign-On (see page 63).

(Optional) Configure Single Sign-On

Note: The option to configure single sign-on settings only appears in the User Console if you have purchased the SSO service.

During partnership configuration for an SSO application, a hosting administrator specifies a *federation attribute* for the partnership. The system uses this attribute to exchange information with the target software resource during single sign-on operations. For example, when configuring an SSO partnership between CA CloudMinder and salesforce.com, a hosting administrator chooses User ID as the federation attribute. The system retrieves this attribute from the database and forwards it in a SAML assertion to salesforce.com to facilitate single sign-on.

Some target software resources require the federation attribute to have a specific format. If this format differs from the format CA CloudMinder uses for the attribute, use the following steps to set the attribute value to the required format. This process is named setting the rule string for the attribute.

Note: Only configure the rule string if the software resource requires that the attribute take a format different from the way it is stored in the CA CloudMinder database.

Follow these steps:

1. In the Create Application screen, click Configure Single Sign On settings for the application.

   The Single Sign On configuration settings appear.

2. Select the Federation User Attribute.

   The attribute that you choose must match the assertion attribute that is indicated in the SSO partnership configuration for this application. If the attribute names do not match, users cannot successfully access this application through SSO. To confirm the assertion attribute name that is indicated in the partnership configuration, refer to your hosting administrator. SSO partnership configuration information is available in the CSP Console.

3. Configure the rule string for the Federation User Attribute.

   The rule string is the format that you want the attribute to take when the system passes it to the target software resource.

   Note: To learn the exact format that is required for this attribute, refer to your hosting administrator, or an administrator at the target software resource.

You have created an application and applied an authentication method. You have also configured single sign-on settings if applicable. You can now include this application in a service so that users can access the application.

Create a Service Using the Service Wizard

Services simplify *entitlement* management. A service bundles together all the entitlements - applications, roles, groups, and attributes - a user needs for a given business role.

For example, all new sales employees need secure access to Salesforce.com and a Salesforce.com account. They need membership in the CA CloudMinder Sales group. They also need specific information added to their user account profiles. An administrator creates a service named Sales Administration that combines the required application, roles, group, and profile attribute information. When an administrator assigns the Sales Administration service to a user, the user becomes a *member* of the service. The user receives the entire set of entitlements that you define in the service.

As an administrator, a key use of services is to give users access to the software resource defined by an application. To give users this access, create a service that includes the application, then make the service available to users.

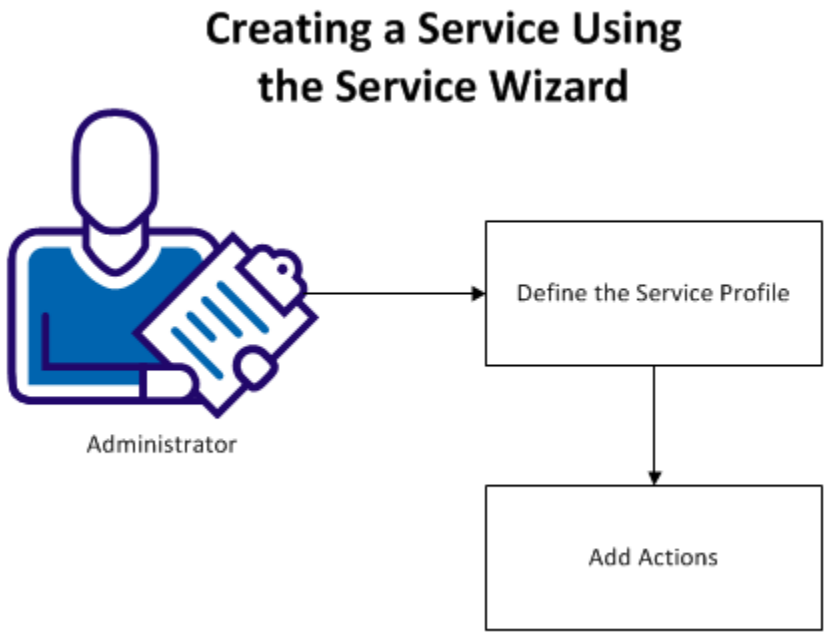
Users access the software resource in the manner that is defined in the application. For example, you can configure the application for Single Sign-on, or you can protect it with your choice of authentication method. Users who access the application through a service receive the benefit of these configurations.

You make the service available various ways. You can assign a service directly to one or more users. You can configure the service so that an application icon appears in the User Console of service members. Users can click the icon to access the application. Users can also request access to a service themselves through the User Console. For more information, see Make a Software Resource Available to Users (see page 68).

Note: As an administrator, you can also give users access to an application by making a link to the application available. For example, you can insert the link into any web page or you can send the link in an email. For more information, see Creating a Service: Service Wizard.

The simplest way to create a service is by using the service wizard.

The following diagram shows the steps to perform, to create a service through the service wizard and make it available to users.



The following topics explain how to create a service using the service wizard:

1. Define the Service Profile (see page 65)
2. Add Actions (see page 67)

Define the Service Profile

On the Profile tab, you define basic characteristics of the service.

Follow these steps:

1. Log in to an account that has service management privileges.

   For example, the default Tenant Administrator role has the appropriate privileges.
2. From the navigation menu, select Services.
3. Click Service Wizard, Create Service.

   The Create Services screen appears.
4. On the Profile tab, enter a name and tag. A tag is a unique identifier for the service.

   Note: Tags can only contain alphanumeric and underscore characters, and cannot start with a number. Once created, a tagname cannot be changed, or reused, even if a service is later deleted.

5. Select Enabled if you want to make the service available to users as soon as you create it.
6. If you want this service to appear in the list of services available for users to request, select Self Subscribing. When Self-Subscribing is enabled, users can request access to this service through the User Console.
7. (Optional) Add one or more categories. Type a category name and click the up arrow to add it to the service.

   Categories add more information to a service. You can use this additional information to facilitate service searches in environments that include a significant number of services.

Add Actions (Service Wizard)

On the Actions tab, you define what entitlements are granted to the user - for example, access to an application, or membership in a role or group - when a user receives the service.

On the Actions tab, from the drop-down menu, select one or more of the following actions:

Application

Service members receive access to the selected application. The system applies group membership or rule string configuration for the application, if necessary. For more information, see Creating an Application (see page 58).

Launch Role

The system adds a link and icon for the application in the User Console of service members. This action need only be added when you want to give users access to an application through this service. Select one of the following options:

Create a new launch role for an application

Select the application that you want service members to be able to access, and enter a name for it. The system adds an icon with this name in the User Console Home page of service members. The system also adds a link with this name in the left-hand navigation pane of the User Console of service members. The icon and the link both launch the application.

Create a new launch role

This action is the same as creating an admin role. If you have task administration privileges, you can add one or more tasks to this role through the Modify Admin Role task. The system then adds a link to these tasks in the left-hand navigation pane of the User Console of service members. For more information, see Admin Roles and Tasks.

Select an existing launch role

Select the launch role that you want service members to be able to access. This role can be a launch role for an application, or for any admin role.

Provisioning Role

Service members receive the selected provisioning role. The system creates an account for the user in the endpoint that is configured in the provisioning role when a user receives the service. For more information, see Creating Roles to Assign Accounts (see page 53).

Group

The system adds service members to the selected group.

Attribute

The system adds the indicated attribute to the user accounts of service members.

The system adds each action that you select to the service. When a user receives access to the service, the system applies each action to the user. For example, the user receives the indicated Launch Role, and access to an application in their User Console Home page.

Important! You can set the order of actions in a service. If you add actions that affect user attributes, and have a provisioning action in the service, order is important. Place actions that affect attributes **before** any provisioning actions in the action order. The Attribute action can affect user attributes. The Application action can also affect user attributes, if a rule string is configured for the application.

Note: Typically, the application and provisioning role you select while creating a service are closely related. The application makes a specific software resource available in the system. The provisioning role creates a user account in the same software resource. Thus, when a user receives the service, they automatically receive access to the software resource through their User Console, and an account in the same software resource.

When you have added all desired actions to the service, click Submit. The system creates a service with the selection actions. When a user receives access to the service, the user receives access to the selected application. The user also receives all roles, groups, and attributes you included in the service.

Make a Software Resource Available to Users

As an administrator, you want to make a software resource that you configured through CA CloudMinder available to users. Depending on the service you purchased, you can configure this software resource for one or more of the following services:

- Single sign-on (SSO)
- Advanced authentication
- Provisioning
- Self-registration

Before you make the resource available to users, confirm that you have performed the following steps, if applicable. (The responsible administrator is indicated in parentheses.)

- For SSO, configure one or more partnerships (hosting administrator)
- Configure and apply an authentication scheme (hosting administrator)
- Create authentication methods (tenant administrator)
- For automatic account creation, create roles provisioning roles (tenant administrator)
- Create an application, optionally enabling self-registration (tenant administrator)
- Create a service (tenant administrator)

When you have completed all configuration, you can make the software resource available to users. You can make the resource available by using one or more of the following methods:

1. Assign the service to a user (see page 70).
2. Allow users to request access to the service.

   In the CA CloudMinder User Console, when the user clicks My Access, then Request & View Access, the user sees a list of services available for their request. The services that appear in this list are those marked "Self Subscribing" service creation.

   When the user requests access, the system assigns the service to the user. The user receives all applications, roles, groups, and attributes that are associated with the service. If the service includes a Launch Role for an application, an icon and a link to the application appear in the User Console Home page.
3. Give users a link to the software resource.

   Give users access to the application by making a link to the application available. For example, you can insert the link into any web page or you can send the link in an email.

   You can use the link from the Administrative UI. From the Federations tab, select Partnership Federation, Partnerships. Select the appropriate partnership from the list to display the partnership configuration. In the SSO section, locate the link labeled *SSO Service URL.* To provide access to the application, use this link.

Assign a Service to a User

You can assign a service directly to an individual user. This user becomes a *member* of the service.

Follow these steps:

1. Navigate to Services, Request & View Access.

   A list of services you can administer appears.

2. Select the service that you want to assign to a user and click Select.

   A list of users that are assigned to the service appears.

3. Click Request Access.

4. Search for a user to whom you want to assign the service.

   To display a list of all users for whom you have administrative privileges, click Search without modifying the search criteria.

5. Select a user and click Select.

   An updated list of users that are assigned to the service appears.

6. Click Save Changes.

   The user receives the specified service. The user receives all applications, roles, groups, and attributes you included in the service.

Chapter 3: SSO using a Third-party IdP and Self-registration

Consumers can have access to an application using credentials from an account with a third-party site. The third-party site acts as an external Identity Provider (IdP) relative to CA CloudMinder.

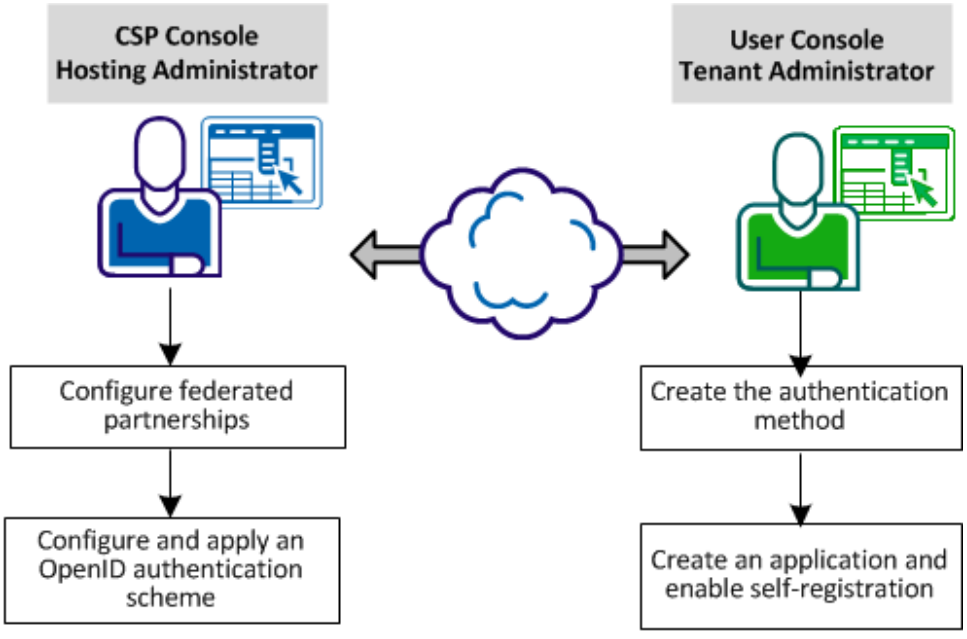
In this scenario, a user accesses a protected resource. The user is shown a page that displays a choice of third-party sites. From the list, the user chooses the third party where they have an account. After successful authentication, the third-party returns the user to the cloud system.

At the cloud system, CA CloudMinder can redirect the user to a self-registration page. The page enables the user to register and establish a user record in the cloud system user directory. After successful registration, CA CloudMinder redirects the user to the requested application.

In this scenario, the following information applies:

- A third party is the IdP that authenticates the user. For example, Facebook.
- CA CloudMinder has two partnership roles:
  - As the SP partner for the third-party IdP.
  - As the IdP that provides the assertion to the SSO application, which is the SP.
- The internal cloud user store is in use.
- SAML 2.0 is the federation profile in use.
- Self-registration is enabled for the application.

The configuration tasks are shown in the following figure:



The following procedures describe each task in detail:

1. Create federated partnerships (see page 72).
2. Configure and apply and authentication scheme (see page 112).
3. Create the authentication method (see page 126).
4. Create an application and enable self-registration (see page 58).

Configure Federated Partnerships

A common SSO scenario is to allow consumers access to an application using credentials from an account at a third-party site. The third-party site acts as an external Identity Provider (IdP) relative to CA CloudMinder.

The following information for the partnership applies:

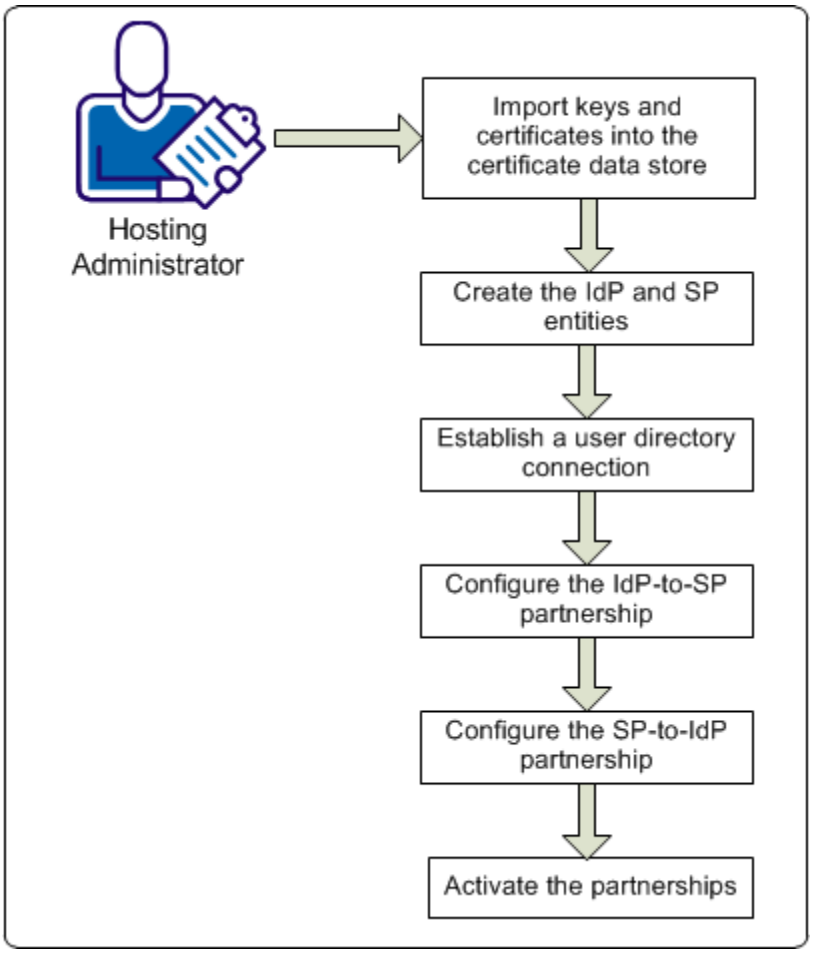
- A third party is the IdP that authenticates the user. For example, Facebook.
- CA CloudMinder has two partnership roles:
  - As the SP in relation to the third-party IdP.
  - As the IdP that provides the assertion to the SSO application, which is the SP.
- SAML 2.0 is the federation profile in use.

Set up two partnerships:

- IdP (third-party) to SP CA CloudMinder
- SP CA CloudMinder to IdP (application)

Note: In many of the procedures, the term *asserting party* refers to the Producer or Identity Provider. The term *relying party* refers to the Consumer, Service Provider, and Resource Partner.

The following figure shows the configuration tasks required for a partnership:



The procedures are detailed in the following topics:

1. Import keys and certficates into the certificate data store (see page 20).
2. Create the IdP and SP entities (see page 26)
3. Establish a user directory connection (see page 34).
4. Configure the IdP-to-SP partnership (see page 97).
5. Configure the Sp-to-IdP partnership. (see page 88)
6. Activate the partnership (see page 111).

Import Keys and Certificates into the Certificate Data Store

Private keys and certificates are required for the following tasks:

- Federation components use private key/certificate pairs for signing, verification, encryption, and decryption of entire assertions, or specific assertion content.
- Federation components employ client certificates for back-channel authentication for artifact single sign-on.
- For establishing SSL connections (SSL server certificates).

Private key/certificate pairs and single certificates for federation functions are stored in the certificate data store (CDS). The certificate data store is collocated with the policy store. All Policy Servers that share a common view into the same policy store have access to the same keys, certificates, CDS-configured certificate revocation lists (CRL), and OCSP responders.

SSL server certificates are stored on the web server where they are installed. SSL server certificates are not stored in the certificate data store.

Each key/certificate pair, client certificate, and trusted certificate in the certificate data store must have a unique alias. The alias allows any private key/certificate pair or single certificate in the certificate store can to be uniquely referenced. The certificate data store can store multiple key/certificate pairs and single certificates. In a federated environment, you can have multiple partners. For multiple partners, you can use a different pair for each partner.

If a signing alias is configured for signing assertions, the assertion generator uses the key that is associated with that alias to sign assertions. If no signing alias is configured, the assertion generator uses the key with the *defaultenterpriseprivatekey* alias to sign assertions. If the assertion generator does not find a default enterprise private key, it uses the first private key it finds to sign assertions.

Important! If you are going to store multiple keys, define the first key that you add with the defaultenterpriseprivatekey alias before adding subsequent keys.

A given Policy Server can sign or sign and verify responses. You can add keys and certificates for signing and validation to the same certificate data store.

You manage the contents of the certificate data store using the Administrative UI.

The following types of key/certificate pairs and single certificates are stored in the certificate data store:

Function	Private Key/Cert Pair	Certificate (public key)	CA Certificates	Client Certificate
Signs assertions, authentication requests, SLO requests and responses	X			
Verifies signed assertions, authentication requests, and SLO requests/responses		X		
Encrypts assertions, Name ID and attributes (SAML 2.0 only)		X		
Decrypts assertions, Name ID and attributes (SAML 2.0)	X			
Serves as a credential for client certificate authentication of the artifact back channel				X
Validates other certificates and certificate revocation lists			X	
Use SSL connections to resolve web services variables			X	

If you do not have a key/certificate pair in the certificate data store, you have two options:

- Import a key/certificate pair from an existing file (.p12 or .pfx).
- Generate a key/certificate pair.

  To generate a new key/certificate pair, request a certificate from a trusted Certificate Authority and then import the signed certificate response that the authority returns.

For more information about key and certificate management, see the *CA SiteMinder Policy Server Configuration Guide.*

Import a Key/Certificate Pair from an Existing File

If you do not have a key/certificate pair in the certificate data store, import one from an existing .p12 or .pfx file.

The Policy Server treats an imported certificate as a trusted certificate. The exceptions are self-signed certificates, which get treated according to the following guidelines:

- The Policy Server identifies a V3 self-signed certificate as a CA certificate. In this case, it treats it as a CA certificate. This behavior occurs even though you initiate the import from the Certificate/Private Key dialog.
- The Policy Server treats the certificate as a trusted certificate when:
  - The Policy Server does not identify a V3 self–signed certificate as a CA.
  - The certificate is a V1 self–signed certificate.

Follow these steps:

1. Log in to the Administrative UI.
2. Select Infrastructure, X509 Certificate Management, Trusted Certificates and Private Keys.
3. Click Import New and follow the wizard.
4. Be aware of the following items as you complete the wizard:
   - You can import a single file with a key and certificate in it or separate key and certificate files. Select the appropriate option button for the file you are using.
   - To import a self-signed certificate as a Certificate Authority certificate, set the Use as CA option button to Yes. The certificate is imported as a CA certificate and is not available for when configuring partnerships (for example, for signing or encryption).

     Otherwise, accept the default No setting to import the certificate as a trusted certificate that is available when configuring partnerships.
   - For a trusted certificate file in DER (binary) format, the file can contain one or more certificate entries. For a trusted certificate file in PEM (base 64) format, one certificate per file is required.
   - If you are using a .p12 file, you are required to fill in a password.
   - For each entry you plan to add to the certificate data store, enter the alias you want to associate with that entry. If you select multiple entries, each requires a unique alias.
5. At the Confirm step, review the information and click Finish.

The key/certificate pair is imported into the certificate data store.

How to Generate a Key/Certificate Pair

If you do not have a key/certificate pair in the certificate data store, you can generate a new key/certificate pair.

Follow these steps:

1. Generate a certificate request and send the request to a trusted Certificate Authority.
2. Import the signed certificate response from the authority.

Generate a Certificate Request

If you do not have a key/certificate pair in the certificate data store, request one from a trusted Certificate Authority. When the CA returns a signed certificate response, import it into the certificate data store.

When you generate a certificate request, the Policy Server generates a private key and a self–signed certificate pair. The Policy Server stores this pair in the certificate data store. Using the generated request, contact a Certificate Authority and fill out the CA certificate request form. Paste the contents of the generated request into the form.

The CA issues a signed certificate response, usually in PKCS #7 format. You can import the signed certificate response into the certificate data store. After the signed certificate response is imported, the existing self–signed certificate entry of the same alias is replaced.

Follow these steps:

1. Log in to the Administrative UI.
2. Select Infrastructure, X509 Certificate Management, Trusted Certificates and Private Keys.
3. Click Request Certificate.
4. Complete the required fields.
5. Click Save.

A file that conforms to the PKCS #10 specification is generated.

The browser prompts you to save or open the file, which contains the certificate request. If you do not save this file (or open it and extract the text), the Policy Server still generates the private key and self–signed certificate pair. To get a new request file for the private key, generate a new certificate signing request using the Generate CSR feature.

Import a Signed Certificate Response

After completing a certificate request and sending it to the Certificate Authority, the Certificate Authority issues a signed certificate response.

Import the signed certificate into the certificate data store to replace the existing self-signed certificate entry of the same alias.

Follow these steps:

1. Select Infrastructure, X509 Certificate Management, Trusted Certificates and Private Keys.
2. In the list, locate the self-signed certificate that you want to update.
3. Select Action, Update Certificate next to the self-signed entry.
4. Browse to the file you want. You can use a:
   - .p7 or .p7b file that contains the signed certificate and the corresponding certificate chain.
   - .cer or .crt file (base64 PEM file) with the signed certificate without the certificate chain.
5. Select the appropriate entry.
6. Review the certificate information and click Finish.

The signed certificate is imported into the certificate data store and the self-signed certificate is replaced.

Generate a New Certificate Signing Request

A certificate signing request (CSR) is a message that you send to a Certificate Authority to apply for a digital identity certificate. After you create a private key, you can generate a CSR. The CSR contains the public key.

You can generate a new CSR for a self-signed or CA-signed private key/certificate pair. The private key always generates an identical CSR without modifying the existing private key. You generate a new request for an existing private key for the following reasons:

- You no longer have the original request that was generated for the private key/self-signed certificate pair.
- You need a new certificate for an expiring one, which requires a new copy of a CSR to submit to a Certificate Authority.

Follow these steps:

1. Log in to the Administrative UI.
2. Select Infrastructure, X509 Certificate Management, Trusted Certificates and Private Keys.
3. Select Action, Generate CSR for the private key entry for which you want a new CSR.

   A file that conforms to the PKCS #10 specification is generated.
4. Save the CSR when prompted.
5. (Optional) If you require a CA-signed certificate, contact a Certificate Authority. Follow the procedure the Certificate Authority requires for submitting a request. Use the PKCS#10 file you saved in the previous step for the request.

After you complete the certificate request process, the Certificate Authority issues a signed certificate response that you import into the certificate data store. The Policy Server replaces the existing certificate entry of the same alias with the newly imported certificate.

Update Certificates in the Certificate Data Store

You can update key/certificate pairs and standalone certificates in the following ways:

- Update an expiring trusted certificate by deleting the existing certificate and importing a new trusted certificate. The new certificate must match the expiring certificate in the certificate data store.
- Update the certificate by importing a signed trusted certificate or a PKCS7-signed response. The new certificate must match the expiring certificate in the certificate data store.
- Update a certificate with a certificate from a PKCS#12 file. The new private key and certificate pair must match the expiring key/certificate pair in the certificate data store.

The new certificate must be valid before the Policy Server can use it to update an expiring certificate. Certificates are updated and become available immediately after they are imported. If the new certificate is not valid, as determined by its validity interval, the Policy Server cannot use the new certificate.

For importing only a trusted certificate, use a file containing the certificate in a PEM or DER encoding. The standard extension for files of these types is *.crt or *.cer. If the file ends in .p12 or .pfx, it is processed as a certificate data store file containing key/certificate pairs. Finally, if a file ends in .p7 or .p7b, it is processed as a signed response file. Anything else is treated as a certificate file, and CA SiteMinder tries to load a certificate from it.

Note: If you update certificates for a federated environment, you do not have to update any federation objects that use the expiring certificates.

Create the IdP and SP Entities

When CA CloudMinder is providing the identity information for the user, it is acting as the local IdP. The business partner, for example, SalesForce.com, is the remote SP.

Each partner in a federation partnership is considered a *federation entity*. Before you establish a partnership, define a local entity that represents the local partner and a remote entity that represents the remote partner.

The two ways to configure a federation entity are:

- Create an entity without using metadata (see page 26).
- Create an entity by importing metadata (see page 31).

Create an Entity without Using Metadata

Create an entity without metadata by using the following process:

1. Indicate an entity type.
2. Configure the specifics about that entity type.
3. Confirm the entity configuration.

Entity Type Choice

The first step in configuring an entity is to establish the entity type and determine the entity role.

To establish the entity type

1. Log in to the Administrative UI.
2. Select Federation, Partnership Federation, Entities.
3. Click Create Entity.

   The Create Entity dialog displays.

   Note: Click Help for a description of fields, controls, and their respective requirements.
4. Select *one* of the following options:

   Local

   Indicates that you are creating an entity that is local to your site.

   Remote

   Indicates that you are configuring an entity that represents the partner at the remote site.

5. Configure the remaining fields:

   New Entity Type

   Select the asserting or relying party.

   SAMLToken Type (WS-FED only)

   Select the token type, which defines the SAML format for the encrypted token that contains user credential information. Choose the Legacy option only if you want the token to comply with the SAML token type for WS-Federation 1.0.

6. Click Next to configure specifics about the entity.

Detailed Local Entity Configuration

After you have specified the entity type, configure the details of the entity. For a local entity, define the following information:

- Identification information about the entity
- Signature and encryption options
- Name ID formats and attributes

Follow these steps:

1. Begin at the Configure Entity step.
2. Complete any required fields for features and services for the local entity type you are configuring.

   Click Help for a description of the fields.

3. Click Next.

   The Confirm dialog is displayed.

Be aware of the following features:

Entity ID and Entity Name Settings

If the Entity ID represents a remote partner, the value must be unique. If the Entity ID represents a local partner, it can be reused on the same system.

The Entity Name identifies an entity object in the policy store. The Entity Name must be a unique value. This value is for internal use only; the remote partner is not aware of this value.

Note: The Entity Name can be the same value as the Entity ID, but do not share the value with other entities at the same site.

Signing and Encryption Features

For signing and encryption features, you must have the appropriate key/certificate entries in the certificate data store. If you do not have the appropriate key/certificate entries, click Import to import a private key/certificate pair from a file on your local system. You can also import trusted certificates.

Note: If you are using SAML 2.0 POST profile, signing assertions is required.

WSFED Attributes (WS-Federation only)

You can specify various service URLs and IDs for WS-Federation entites to communicate.

Name ID Formats

You can indicate the identifier types that the federated entity supports.

Assertion Attribute Configuration (asserting partners only)

You can configure the asserting party to include specific assertion attributes when it generates an assertion. The recommended method is to define these attributes at the entity level. The entity serves as a template for the partnership so any assertion attributes you define for the entity get propagated to the partnership. The benefit of defining assertion attributes at the entity is that it enables you to use an entity in more than one partnership.

If you want to add or remove assertion attributes for the partnership, make such modifications at the partnership level, not at the entity level.

Detailed Remote Entity Configuration

After you have specified the entity type, configure the details of the entity. For a remote entity type, define the following information:

- Identification information about the entity
- Signature and encryption options
- NameID and attribute information

Follow these steps:

1. Begin at the Configure Entity step.
2. Specify the Assertion Consumer Service URL. Examples:
   - If the SP is a site such as Google, the URL can be similar to:

     https://www.google.com/a/example.com/acs
   - If the SP is a site such as Salesforce.com, the URL can be similar to:

     https://login.salesforce.com/?saml=EK05LGnm40H7
   - If the SP is another business partner, the URL can be similar to:

     http://myserver.forwardinc.com:9080/samlsp/acs

3. Complete any other required fields for features and services for the remote entity type.

   Click Help for the field descriptions.

4. Click Next.

   The Confirm dialog is displayed.

Be aware of the following features:

Entity ID and Entity Name Settings

If the Entity ID represents a remote partner, the value must be unique. If the Entity ID represents a local partner, it can be reused on the same system.

The Entity Name identifies an entity object in the policy store. The Entity Name must be a unique value. This value is for internal use only; the remote partner is not aware of this value.

Note: The Entity Name can be the same value as the Entity ID, but do not share the value with other entities at the same site.

Signing and Encryption Features

For signing and encryption features, you must have the appropriate key/certificate entries in the certificate data store. If you do not have the appropriate key/certificate entries, click Import to import a private key/certificate pair from a file on your local system. You can also import trusted certificates.

Note: If you are using SAML 2.0 POST profile, signing assertions is required.

WSFED Attributes (WS-Federation only)

You can specify various service URLs and IDs for WS-Federation entites to communicate.

Name ID Formats

You can indicate the identifier types that the federated entity supports.

Assertion Attribute Configuration (asserting partners only)

You can configure the asserting party to include specific assertion attributes when it generates an assertion. The recommended method is to define these attributes at the entity level. The entity serves as a template for the partnership so any assertion attributes you define for the entity get propagated to the partnership. The benefit of defining assertion attributes at the entity is that it enables you to use an entity in more than one partnership.

If you want to add or remove assertion attributes for the partnership, make such modifications at the partnership level, not at the entity level.

Confirm the Entity Configuration

Review the entity configuration before saving it.

Follow these steps:

1. Review the settings in the entity dialog.
2. Click Back to modify any settings from this dialog.
3. Click Finish when you are satisfied with the configuration.

A new entity is configured.

Editing Entities from the Partnership

You can click Get Updates next to the local and remote entity fields to update information about the entity. When you select Get Updates, the system asks to pull in the latest information from the entity.

After confirmation, the partnership you are editing is refreshed with the latest entity information. Changes are saved when you complete the partnership wizard. If you do not confirm the update, the partnership configuration remains the same.

The Entity Name identifies an entity object for in the policy store. The Entity Name must be the unique identifier because the product uses this value internally to distinguish an entity. This value is not used externally and the remote partner is not aware of this value.

If the Entity ID represents a remote partner, the value must be unique. If the Entity ID represents a local partner, it can be reused on the same system.

Note: The Entity Name can be the same value as the Entity ID, but do not share the value with any other entity.

An entity is a key component of a federation partnership. Changing an entity alters the partnership significantly; therefore, the Administrative UI does not let you replace an entity after it is in a partnership. To replace an entity, create a partnership.

To provide some flexibility within partnership configuration, you can change an entity ID because it does not identify the entity uniquely. Changing the entity ID at the partnership level does not link the partnership to another entity. The original entity in the partnership does not change. Modifications to an entity are a one-way propagation from the entity to the partnership. A change to the entity ID at the partnership does not get propagated back to the original entity.

Regard entity configurations as templates. Partnerships are created based on the entity templates so changing the partnership does not change the original entity template.

Create an Entity by Importing Metadata

You can import data from a metadata file to create a federation entity. Importing the metadata reduces the amount of configuration for creating a partnership.

You can use metadata in the following ways:

- Import data from a remote partner to create a new remote entity.
- Import data from a remote partner to update an existing remote entity.
- Import data from a local entity to create a new local entity.

  This option can be useful to facilitate a migration from another federation product.

  Note: Federation does not support metadata imports to update or restore an existing partnership and local entity. To update an existing local entity, edit the entity and modify the settings that you want to change. You can import metadata only to create a *new* local entity.

The process for creating a metadata-based entity is as follows:

1. Select a metadata file for configuring a new entity.
2. Select an entity entry from the metadata file. The file can include several entities, but one entity per file is recommended.
3. (Optional) Select the certificates to import into the certificate data store. The certificates must be in the metadata file.

   These certificates can be used for authentication request verification, single logout response verification (SAML 2.0), and encryption (SAML 2.0).
4. Confirm the entity configuration.

Details about these steps are described in the next sections.

Metadata File Selection

The first step to create an entity from metadata is to select the metadata file.

Follow these steps:

1. Log in to the Administrative UI.
2. Select Federation, Partnership Federation, Entities.
3. Click Import Metadata.

   The Import Metadata dialog opens.

   Click Help for the field descriptions.
4. Browse for the metadata file you want to use to create the entity.

5. Select whether to create a new local or remote entity, or update an existing remote entity.

   Note: The Policy Server does not support metadata imports to update an existing partnership and local entity. You can only create a new local entity. To update an existing local entity, edit the entity and modify the settings that you want to change. You can update the existing remote entities or you can create new remote entities.

6. Click Next to select entities from the file.

If you select a metadata file with expired entries, the next dialog that the UI displays contains a section listing the expired entries. You cannot select these expired entries; they are displayed for your reference. If all entities in a metadata file are expired, no entities are displayed. In this case, upload a new document.

Select an Entity to Import

This procedure assumes that you have already selected a metadata file to create an entity. Select the entity from the file.

Follow these steps:

1. Specify a name for the new entity in the Select Entity Defined in File dialog.

   If you are doing a local import to create an entity, define the partnership name.

2. Click on the option button to select the entity.

3. Click Next.

   The Import Certificates dialog displays if importing metadata for a remote entity and the document includes certificate data.

   If the metadata file that you imported contains certificate entries, you can import these entries.

Certificate Imports

To verify signed assertions, import certificates if the metadata includes them. If the metadata does not include certificates, skit this step and go to the Confirm step.

Follow these steps:

1. From the Import Certificates step, select the certificate entry or entries from the metadata file that you want to import.

   If you select a certificate file with invalid entries, the next dialog contains a section listing the expired entries. You cannot select these expired entries. They are displayed for your reference. If all entries in the file are invalid, the import wizard skips the certificate selection step.

   Specify a unique alias for each entry that you chose.

2. Click Next

   The Confirm dialog displays showing a table of entries.

   You can select two entries from a metadata file that have the same certificate. For SAML 1.1 metadata, every entry shows Signing as the usage for the certificate because SAML 1.1 does not encrypt data.

   For SAML 2.0, each entry can show a different usage for the certificate, for example, one for signing, one for encryption. When you get to the Confirm step, the window shows a table with a single certificate entry. The certificate usage is listed as Signing and Encryption. This entry is the combination of the two entries you chose previously. This entry also uses the first alias that you specified for the certificate entry you selected.

   This situation occurs only if the same certificate was listed in the metadata file for both uses. If the file contains two separate certificates, the confirmation step shows both entries in the table.

   For example, you select two entries from the metadata file and you do not realize they are the same certificate. The first usage is Signing and you assign it the alias **cert1.** The second usage is Encryption and you assign it the alias **cert2**. When you confirm the import, you see a table titled Selected Certificate Data with an entry similar to the following entry:

Alias	**Issued To**	**Usage**
cert1	Jane Doe	Signing and Encryption

   If no usage is specified in the metadata file, then the usage defaults to Signing and Encryption.

3. Click Next to finish the configuration.

Confirm the Entity Configuration

Review the entity configuration before saving it.

Follow these steps:

1. Review the settings in the entity dialog.
2. Click Back to modify any settings from this dialog.
3. Click Finish when you are satisfied with the configuration.

A new entity is configured.

Establish a User Directory Connection

Partnership federation looks up entries in a user directory to verify identities and retrieve user attributes for a given principal. At the asserting party, the federation partner generates assertions for the appropriate users, and authenticates each user against a user directory. At the relying party, the federation partner extracts the necessary information from an assertion and looks in the user directory for the appropriate user record.

Configure connections to existing user directories by selecting Infrastructure, Directory, User Directories in the Administrative UI. You are only establishing a connection to an existing user directory. You are not configuring a new user directory.

Note: To use an ODBC database in your federated configuration, set up the SQL query scheme and valid SQL queries before selecting an ODBC database as a user directory.

Configure connections to more than one directory if necessary. The directories do not have to be the same type.

For detailed information about user directories, see the *Policy Server Configuration Guide.*

For deployments that use the internal cloud user directory, connect to the internal user directory.

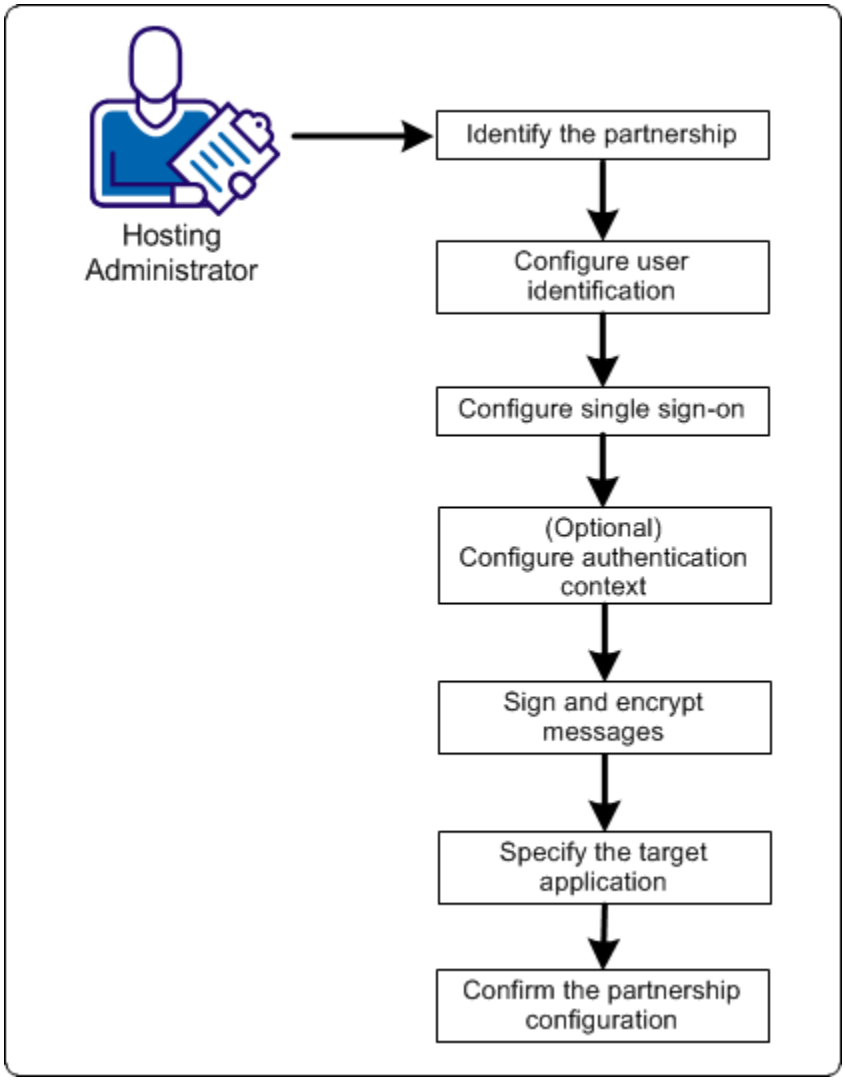
Create the Local SP-to-Remote IdP Partnership

Set up a partnership where CA CloudMinder is the local SP and the third-party is the remote IdP. In this scenario, when a user requests access to an application, the system presents a page from which the user can choose a site to use for login. The user can log in directly if they have an account at the tenant.

This scenario assumes that a consumer application, such as a rewards programs, is the target application. It is unlikely that the user is known to CA CloudMinder. Therefore, the user selects a third-party where they have an account.

The purpose of the SP-to-IdP partnership is to direct the user to the appropriate third-party IdP. The third-party IdP generates an assertion based on the account information it has for the user after that user logs in.

The following figure shows the configuration tasks for the local SP-to-Remote IdP partnership:



Identify the Partnership

Follow these steps:

1. Select Federation, Partnership Federation, Partnerships.
2. Click Create Partnership.
3. Select SAML2 SP -> IdP.

   Selecting this option indicates that you are the local SP and that the IdP is a remote partner.

   You come to the first step in the partnership wizard.

4. Complete the following fields

   Partnership Name

   Local SP

   Select the local SP. Example: cloudhost.ca.com.

   Remote IdP

   Select the remote ID. For example, Facebook.com

   Skew Time (Seconds)

   Accept the default

   The skew time is the difference between the system time on the local system and the system time on the remote system. Usually, the inaccuracy of system clocks causes this condition. Determine the skew time number by subtracting the number of seconds from the current time.

   The system uses the skew time and the SSO validity duration to determine how long an assertion is valid.

5. Move the cloud host directory from the Available Directories list to the Selected Directories list.

   If you configure only one user directory, that directory is automatically placed in the Selected Directories list.

6. Click Next to go to the Federation User step.

Note: If you are editing a partnership, you can click Get Updates next to this field to update the entity information. The latest information from the entity configuration is propagated to the partnership. However, if you edit the entity information directly from the partnership, the changes do not get propagated back to the individual entity configuration.

Configure User Identification at the Relying Party

Configure user identification so the relying party has a method of locating a user in the local user directory.

Follow these steps:

1. Select one of the following attributes for disambiguation:

   - Name ID
   - An attribute from a previously populated drop-down list

     If the remote asserting entity was created based on metadata that contained attributes, the list is populated.

- An attribute you enter.

  This option is most likely used when metadata is not available and the remote asserting entity does not include any attributes.

- An Xpath query

Click Help for the field descriptions,

2. (Optional—SAML 2.0 only) Select Allow IDP to create user identifier.

   This attribute instructs the asserting party to generate a new value for the NameID, if this feature is enabled at the asserting party. The Name ID Format entry at the asserting party must be a persistent identifier.

3. (Optional—SAML 2.0 only) Select Query parameter overrides identifier.

   This setting lets the relying party send an AllowCreate query parameter to override the value of the AllowCreate attribute configured in the authentication request. Using the query parameter instead of the identifier lets you change the value of the AllowCreate attribute without altering the partnership configuration.

   Note: For the Identity Provider to honor this query parameter setting, select the Allow IDP to create user identifier check box.

4. Specify a directory search specification for each directory listed. Two examples of search specifications are:

   LDAP Example

   uid=%s

   ODBC Example

   name=%s

5. Click Next to continue with the partnership configuration.

Single Sign-on Configuration (Relying Party)

To configure single sign-on at the relying party, specify the SAML binding and the other related SSO settings.

At the relying party, the system uses the skew time for the partnership to determine whether the assertion it receives is valid. To understand how the system uses the configured skew time, read more about assertion validity (see page 38).

The procedure that follows offers the basic steps to enable single sign-on. Details about all the configurable features in the sign-on dialog are described in subsequent topics and in the Administrative UI help.

Follow these steps:

1. Begin at the appropriate step in the partnership wizard.

   SAML 1.1

   Single Sign-On

   SAML 2.0

   SSO and SLO

   WS-Federation

   Single Sign-On and Sign-Out

2. Configure the settings in the SSO section of the dialog. These settings let you control the single sign-on binding.

   Click Help for the field descriptions.

   For SAML, configure the HTTP-Artifact or the HTTP-POST profile. If the relying party initiates single sign-on, it includes a query parameter in the request. This query parameter indicates the SSO binding to use. If no binding is specified, the default is POST. If the asserting party initiates single sign-on, the asserting party indicates the binding in use for that particular transaction.

3. (Optional). For SAML 2.0, you can configure these settings:

   - Remote SSO Service URLs
   - Remote SOAP Artifact URLs
   - Initiation of single sign-on from which partner

     If a third-party IdP is authenticating a consumer user with no user record at the host, SSO is initiated at the SP.

   - User consent requirement

4. If you select the HTTP-Artifact profile, configure the authentication method for the back channel in the Back Channel section of the dialog.
5. For the remaining settings, accept the defaults.

The basic settings for single sign-on are complete. Other settings are available for SSO. Click Help for the field descriptions.

Configure Authentication Context Processing (Optional)

The *authentication context* indicates how a user authenticated at an Identity Provider. The Identity Provider includes the authentication context in an assertion at the request of a Service Provider or based on configuration at the Identity Provider. A Service Provider can require information about the authentication process to establish a level of confidence in the assertion before granting access to resources.

Requesting the Authentication Context

To request the authentication context, the CA SiteMinder Service Provider must include the <RequestedAuthnContext> element in the authentication request to the Identity Provider. The Service Provider, puts this element is in the request based on a configuration setting in the SP->IdP partnership.

Obtaining the Authentication Context

A CA SiteMinder Identity Provider obtains the authentication context in *one* of two ways:

- You specify a static AuthnContext URI in the IdP->SP partnership configuration.

  If the federated partner is a CA SiteMinder Service Provider that does not support AuthnContext requests, manually enter a URI in the Administrative UI.

- The AuthnContext URI is determined dynamically using a configured authentication context template.

  The Policy Server maps the authentication context URIs to Policy Server-defined authentication levels. The authentication levels indicate the strength of an authentication context for an established user session. The levels enable the authentication context to be derived from the user session at the Identity Provider.

When the Identity Provider receives a request, it compares the value of the <RequestedAuthnContext> element to the authentication context. The comparison is based on a comparison value in the request from the Service Provider. If the comparison is successful, the Identity Provider includes the authentication contexts in the assertion that it returns to the Service Provider. If validation is configured at the Service Provider, the Service Provider validates the incoming authentication context with the value it requested.

This feature is optional. You can skip this step and navigate to Signature and Encryption.

Enable Signature Processing at the Local SP

The Signature and Encryption step in the partnership wizard lets you define how the Policy Server uses private keys and certificates to do the following tasks:

- Verify SAML assertions signatures and assertion responses and sign authentication requests.

  Note: For SAML 2.0 POST binding, the IdP is required to sign assertions.

- Sign single logout responses and requests (HTTP-Redirect and SOAP bindings).

There can be multiple private keys and certificates in the certificate data store. If you have multiple federated partners, you can use a different key pair for each partner.

Note: If the system is operating in FIPS_COMPAT or FIPS_MIGRATE mode, all certificate and key entries are available from the pull-down list. If the system is operating in FIPS-Only mode, only FIPS-approved certificate and key entries are available.

To configure signing options

1. Begin by selecting the Signature and Encryption step in the partnership wizard.
2. In the Signature section, select an alias from the certificate data store for the Signing Private Key Alias field. If there is no private key in the database, click Import to import one. Or, click Generate to create a key pair and generate a certificate request.

   By completing this field, you are indicating which private key the relying party uses to sign authentication requests and single logout requests and responses.

   Note: Click Help for a description of fields, controls, and their respective requirements.
3. Select the hash algorithm for digital signing in the Signing Algorithm field. The SP signs authentication requests and SLO-SOAP messages with the specified algorithm.

   Select the algorithm that best suits your application.

   RSAwithSHA256 is more secure than RSAwithSHA1 due to the greater number of bits used in the resulting cryptographic hash value.

   CA SiteMinder uses the algorithm that you select for all signing functions.
4. Select an alias from the certificate data store for the Verification Certificate Alias field.

   By completing this field, you are indicating which certificate the relying party uses to verify signed assertions or single logout requests and responses. If there is no certificate in the database, click Import to import one.

5. (Optional) For the SP to sign all authentication requests, select the Sign Authentication Requests. If the remote asserting party requires the authentication requests to be signed, check this option.

   Activate a partnership for all configuration changes to take effect and for the partnership to become available for use. Restarting the services is not sufficient.

If you are using CA SiteMinder in a test environment, you can disable signature processing to simplify testing. Click the Disable Signature Processing check box to disable the feature.

Important! Enable signature processing in a SAML 2.0 production environment.

Enable Encryption Processing at the Local SP (Optional)

Specify the Target Application

From the Application Integration step, the Target Application section is where you define how a user is redirected to the target application. The redirection method that you select depends on the type of data you want to pass with the user to the target application.

Follow these steps:

1. Navigate to the Application Integration step in the partnership wizard.
2. In the Redirect Mode field, select a redirection method. Consider the following information:
   - If you select Cookie Data, you can URL-encode attribute data in the cookie by selecting the URL Encode Attribute Cookie Data check box.
   - If you select the open-format cookie, configure the additional required settings. You can also enable optional settings.

     If the relying party receives an assertion with multiple attribute values, CA SiteMinder passes all values to the target application in the cookie.

- If you select one of the FIPS-compatible algorithms (AES algorithms), use a CA SiteMinder® Federation SDK to generate the open-format cookie. If you use the .NET SDK, use only the AES128/CBC/PKCS5Padding encryption algorithm.

  The target application must use the same language as the SDK that creates the cookie. If you are using the CA SiteMinder® Federation Java SDK, the application must be in Java. If you are using the .NET SDK, the application must support .NET.

- If you select HTTP Headers as the redirect mode, the system can deliver multiple attribute values in a single header. Separate each attribute value with a comma.

  Learn more about using HTTP Headers as the redirect mode and how to protect the headers.

Note: Click Help for a description of fields, controls, and their respective requirements.

3. Enter the URL of the target application in the Target field.

   If a proxy sits in front of the server with the target resource, enter the URL for the proxy host. The proxy handles all federation requests locally. The proxy host can be any system that sits in front of the target server. The proxy host can also be CA SiteMinder itself, provided CA SiteMinder is being accessed directly from the Internet. Ultimately, when operating with a proxy, the URL you specify as the target must go through CA SiteMinder. For example, if the base URL is fed.demo.com and the backend server resource is mytarget/target.jsp, the value for this field is http://fed.demo.com:5555/mytarget/target.jsp.

   For SAML 2.0, you can leave this field blank if you override it with the RelayState query parameter. The RelayState query parameter can part of the URL that triggers single sign-on. To enable this override, select the Relay state overrides target check box.

Setting up redirection to the target is complete.

Partnership Confirmation

Review the partnership configuration before saving it.

Follow these steps:

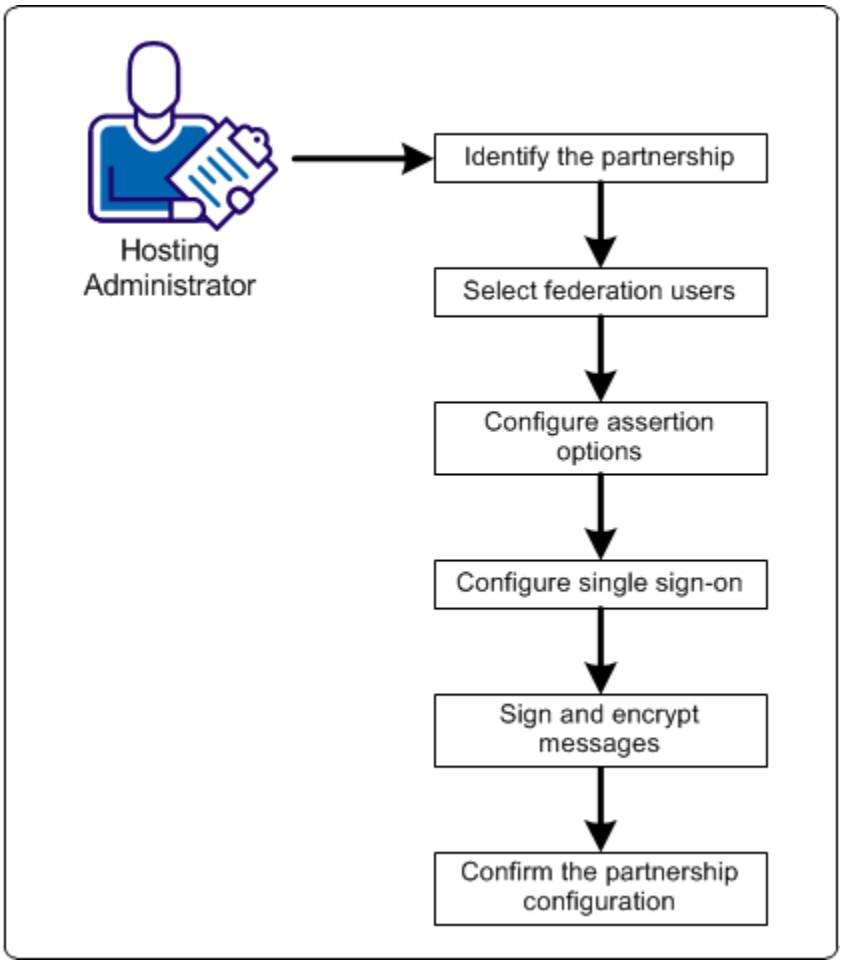
1. Review the settings in the Confirm step of the Partnership wizard.
2. Click Modify in each group box to change any settings.
3. Click Finish when you are satisfied with the configuration.

The partnership configuration is complete.

Create the Local IdP to Remote SP Partnership

Set up an IdP-to-SP partnership between CA CloudMinder (IdP) and the business partner (SP). This partnership is necessary so that CA CloudMinder can pass the assertion that it receives from the third-party IdP to the SP. Using the assertion, the SP authorizes the user for access to the requested application.

The following figure shows the configuration tasks for an IdP-to-SP partnership:



Identify the Partnership

Follow these steps:

1. Select Federation, Partnership Federation, Partnerships.
2. Click Create Partnership.

3. Select SAML2 SP -> IdP.

   Selecting this option indicates that you are the local SP and that the IdP is a remote partner.

   You come to the first step in the partnership wizard.

4. Complete the following fields

   Partnership Name

   Local SP

   Select the local SP. Example: cloudhost.ca.com.

   Remote IdP

   Select the remote ID. For example, Facebook.com

   Skew Time (Seconds)

   Accept the default

   The skew time is the difference between the system time on the local system and the system time on the remote system. Usually, the inaccuracy of system clocks causes this condition. Determine the skew time number by subtracting the number of seconds from the current time.

   The system uses the skew time and the SSO validity duration to determine how long an assertion is valid.

5. Move the cloud host directory from the Available Directories list to the Selected Directories list.

   If you configure only one user directory, that directory is automatically placed in the Selected Directories list.

6. Click Next to go to the Federation User step.

Note: If you are editing a partnership, you can click Get Updates next to this field to update the entity information. The latest information from the entity configuration is propagated to the partnership. However, if you edit the entity information directly from the partnership, the changes do not get propagated back to the individual entity configuration.

Select Federation Users

The Federation Users dialog is the second step in the partnership wizard when the local entity is the asserting party. This step lets you specify which users are authorized to access target resources at the remote site.

Follow these steps:

Note: Click Help for a description of fields, controls, and their respective requirements.

1. Select a user directory from the list in the Directory column of the table of the Federated Users group box.

   The pull-down list consists of one or more directory entries, depending on the number of directories you specified in the previous dialog.

2. Select the user class in the User Class column. This entry specifies a category of individual users or groups of users that can be authenticated. The options for this field depend on the type of user directory (LDAP or ODBC). Refer to the User Class tables for an explanation and example of each user class.

3. Enter a name or filter in the User Name/Filter By column. The value in this column lets the system locate the user or user group from which to authenticate federated users. This entry is dependent on the value you select for the User Class column. For examples of names and filters, see the tables at the end of this procedure.

4. (Optional) You can select Exclude for an entry to indicate that you want to exclude this user class. The default is to include all users in the directory.

   Note: An exclude criteria always takes precedence over an include criteria in case the two criteria conflict.

5. (Optional) Click Add Row to specify another user class for the same directory or another user directory.

The selection of users is complete.

Examples of User Class Entries

LDAP Examples

Use the LDAP filter syntax when specifying entries.

User Class	Valid Entry
User	Distinguished name of a user. Example: uid=user1,ou=People,dc=example,dc=com
Group	Group chosen from the list. Example: ou=Sales,dc=example,dc=com

User Class	Valid Entry
Organization Unit	Organizational unit chosen from the list. Example: ou=People,dc=example,dc=com
Filter User Property	LDAP filter. The current user is the starting point for the search. **Example 1:** mail=user@example.com **Example 2:** (|(mail=*@.example.com)(memberOf=cn=Employees,ou=Groups,dc=example,dc=com))
Filter Group Property	LDAP filter. The current user gets authorized if they are a member of one of the groups matching the filter. The objectclasses for groups as configured in the SiteMinder registry are combined with the filter. **Example 1:** To authorize users that are members of a group with a business category of "CA Support", enter: businessCategory=CA Support **Example 2:** To authorize users that are members of a group with a description containing "Administrator" and a business category of "Administration", enter: (|(description=*Administrator*)(businessCategory=Administration)) **Note:** Not all attributes of a group work as a search criterion.
Filter OU Property	LDAP filter. The current user gets authorized if they belong to an organizational unit that matches the filter. The objectclasses for organizational units as configured in the SiteMinder registry are combined with the filter. **Example 1:** To authorize users within an organizational unit with a postal code of "12345", enter: postalCode=12345 **Example 2:** To authorize users in an organizational unit with a preferred delivery method ending with "phone" and a locality of "London", enter: (|(preferredDeliveryMethod=*phone)(l=London))

User Class	Valid Entry
Filter Any	LDAP filter. The current user gets authorized if they match the filter. **Example 1:** To authorize users with a department of "CA Support", enter: department=CA Support **Example 2:** To authorize users who are members of the group "Administrators" and have a department number of "123" or "789", enter: (&(memberof=cn=Administrators,ou=Groups,dc=example,dc=com)(|(departmentNumber=123)(departmentNumber=789)))

ODBC Examples

Use the SQL syntax when specifying queries.

User Class	Valid Entry
User	Value of the Name column for a user. The current user gets authorized if they match the entry. Example: user1
Group	Value of the Name column of a user group. The current user gets authorized if they are a member of the group that matches the query. Example: Administrators
Query	A SQL SELECT statement. The current user gets authorized if they match the query. **Example 1:** With a userid of user1: Entry: SELECT * FROM SmUser Resulting query: SELECT * FROM SmUser WHERE Name = 'user1' **Example 2:** With a userid of user1: Entry: SELECT * FROM SmUser WHERE Status LIKE 'Active%' Resulting query: SELECT * FROM SmUser WHERE Status LIKE 'Active%' AND Name = 'user1' **Example 3:** With a userid of user1: Entry: SELECT * FROM SmUser WHERE Location IN ('London', 'Paris') Resulting query: SELECT * FROM SmUser WHERE Location IN ('London', 'Paris') AND Name = 'user1'

Configure Assertion Options

Configure assertion options in the Assertion Configuration step of the partnership wizard.

Follow these steps:

1. Configure the settings in the Name ID section.

   The relying party uses these values to know how to interpret the value that is passed in the assertion.

   Based on the value of the NameID Type, complete one of the following tasks:

   - If you selected Static or User Attribute for the Name ID type, complete the Value field.
   - If you selected the DN Attribute for the Name ID type, complete the Value and the DN specification fields.

   Note: Click Help for a description of fields, controls, and their respective requirements.

2. (Optional - SAML 2.0 only) Select Allow Creation of User Identifier so the asserting party can create a value for the NameID. For this feature to work, the AuthnRequest from the relying party must include an AllowCreate attribute.

   Note: If you select this option, the value of the Name ID Format value must be Persistent Identifier.

3. (Optional) Click Add Row in the Assertion Attributes table to specify one or more attributes for inclusion in the assertion. Optionally, you can encrypt the attribute.

   Click Help for detailed information about the columns in the attribute table.

   Note: For attributes from an LDAP user store, you can add multivalued user attributes to an assertion. The Help describes how to specify multivalued user attributes.

4. (Optional) If you have written an assertion generator plug-in using the CA SiteMinder® Federation Java SDK, complete the fields in the Assertion Generator Plug-in section.

   To write a plug-in, see the *Programming Guide for Federation Manager Java SDK*.

5. Click Next to continue with partnership configuration.

Single Sign-on Configuration (Asserting Party)

Configure single sign-on at the asserting party to specify how the asserting party delivers an assertion to a relying party.

Follow these steps:

1. Begin at the appropriate step in the partnership wizard.

   SAML 1.1 and WS-FED

   Single Sign-On

   SAML 2.0

   SSO and SLO

   Any values that are defined during the creation or import of the remote relying party are filled in.

   Note: Click Help for a description of fields, controls, and their respective requirements.

2. In the Authentication section, configure the following fields so CA CloudMinder can act as the IdP

   Authentication Mode

   Delegated

   Delegated Authentication Type

   Cloud

   Delegated Authentication URL

   Enter the URL of the system authenticating the user requesting a resource. Use the following syntax for the delegated URL:

   http://*cloud_system:port*/chs/login/*tenant_name/application_name*

   The *cloud_system* is the system where the user console is installed.

   Example URL:

   http://cmserver.fowardinc.com:832/chs/login/tenant1/confidential_app

   Configure AuthnContext

   Use Predefined Authentication Class

   Authentication Class field

   Supply a static URI for SAML 1.1, SAML 2.0, and WS-FED.

   Additionally, for SAML 2.0 only, the system can automatically detect an authentication class. The URI is placed in the AuthnContextClassRef element in the assertion to describe how a user is authenticated.

3. Complete the fields in the SSO section to determine how single sign-on operates. These settings let you control the following features:

   - Single sign-on binding
   - Assertion validity

     The SSO Validity Duration and the Skew Time determine when the assertion is valid. Read the information about assertion validity (see page 104) to understand how these settings work together.

   For SAML 2.0, you can configure these features:

   - Initiation of single sign-on from which partner
   - SP session validity
   - SP session duration
   - User consent to share identity information with the SP

   Note: Click Help for a description of fields, controls, and their respective requirements.

4. Specify the URL for the Remote Assertion Consumer Service. This service is the service at the relying party that processes received assertions.

   Your partner needs to supply this URL to you.

5. If you selected HTTP-Artifact, configure the back channel settings (see page 39).

Assertion Validity for Single Sign-on

For single sign-on, the values of the Skew Time and the SSO Validity Duration determine how long an assertion is valid. The Policy Server applies the skew time to the generation and consumption of assertions. In the assertion document, the NotBefore and NotOnOrAfter values represent the beginning and end of the validity interval.

At the asserting party, the Policy Server sets the assertion validity. The Policy Server determines the beginning of the validity interval by taking the system time when the assertion is generated. The software sets the IssueInstant value in the assertion from this time. The Policy Server then subtracts the skew time value from the IssueInstant value. The resulting time becomes the NotBefore value.

NotBefore=IssueInstant - Skew Time

To determine the end of the validity interval, the Policy Server adds the Validity Duration value and the skew time to the IssueInstant value. The resulting time becomes the NotOnOrAfter value.

NotOnOrAfter=Validity Duration + Skew Time + IssueInstant

Times are relative to GMT.

For example, an assertion is generated at the asserting party at 1:00 GMT. The skew time is 30 seconds and the validity duration is 60 seconds, making the assertion validity interval between 12:59:30 GMT and 1:01:30 GMT. This interval begins 30 seconds before the time the assertion was generated and ends 90 seconds afterward.

At the relying party, the Policy Server performs the same calculations as it does at the asserting party to determine if the assertion it receives is valid.

Calculating Assertion Validity when CA SiteMinder is at Both Sides of the Partnership

The total time the assertion is valid is the sum of the SSO validity duration plus two times the skew time. The equation is:

Assertion Validity = 2x Skew Time (asserting party) + SSO Validity Duration + 2x Skew Time (relying party)

The initial part of the equation (2 x Skew Time + SSO Validity Duration) represents the validity window at the asserting party. The second part of the equation (2 x Skew Time) represents the skew time of the system clock at the relying party. You multiply by 2 because you are accounting for the NotBefore and the NotOnOrAfter ends of the validity window.

Note: For the Policy Server, the SSO Validity Duration is only set at the asserting party.

Example

Asserting Party

The values at the asserting party are as follows:

IssueInstant=5:00PM

SSO Validity Duration=60 seconds

Skew Time = 60 seconds

NotBefore = 4:59PM

NotOnOrAfter=5:02PM

Relying Party

The relying party takes the NotBefore and NotOnOrAfter values that is receives in the assertion then applies its skew time to calculate new values.

Skew Time = 180 seconds (3 minutes)

NotBefore = 4:56PM

NotOnOrAfter=5:05PM

Based on these values, the calculation for the total assertion validity window is:

120 seconds (2x60) + 60 seconds + 360 seconds (2x180) = 540 seconds (9 minutes).

Back Channel Authentication for Artifact SSO

Artifact single sign-on requires the relying party to send an artifact to the asserting party to retrieve the assertion. The asserting party uses the artifact to retrieve the correct assertion and returns the assertion to the relying party over a back channel.

You can require an entity to authenticate to access the back channel. The back channel can also be secured using SSL, though SSL is not required.

Securing the back channel using SSL involves:

1. Enabling SSL.

   SSL is not required for Basic authentication but you can use Basic over SSL. SSL is required for Client Cert authentication.

2. Configuring an incoming or outgoing back channel for the SAML 2.0 communication exchange. The direction you configure depends on the role of the local entity.

   Configuring separate channels is supported only for SAML 2.0. The back channel configuration for SAML 1.1 artifact single sign-on uses a single configuration for each partnership. CA SiteMinder uses the correct direction automatically (incoming for a local producer and outgoing for a local consumer).

Select which direction to configure for SAML 2.0 single sign-on based on the entity you are configuring.

- The local asserting party uses the incoming channel.
- The local relying party uses the outgoing channel.

Note: You can configure an incoming and outgoing back channel; however, a channel can have only one configuration. If two services use the same channel, these two services use the same back channel configuration. For example, if the incoming channel for a local asserting party supports HTTP-Artifact SSO and SLO over SOAP, these two services must use the same back channel configuration.

3. Choosing the type of authentication for the relying party to gain access across the protected back channel. The authentication method applies per channel (incoming or outgoing).

   The options for back channel authentication are:

   - Basic
   - Client Cert
   - NoAuth

   The Administrative UI help describes these options in detail.

   Important! The authentication method for the incoming back channel must match the authentication method for the outgoing back channel on the other side of the partnership. Agreeing on the choice of authentication method is handled in an out of band communication.

Configure the HTTP-Artifact Back Channel

Protect the HTTP-artifact back channel across which the asserting party sends the assertion to the relying party.

Consider the following limitation:

You cannot use client certificate authentication with the following web servers running ServletExec:

- IIS web servers at a CA SiteMinder producer/Identity Provider because of a limitation in IIS.
- SunOne/Sun Java Server web servers at a CA SiteMinder producer/Identity Provider because of a documented limitation in ServletExec.

Follow these steps:

1. Begin at the Back Channel section in the Single Sign-on or the SSO and SLO step of the partnership wizard.
2. Select HTTP-Artifact in the SSO section.

   The Authentication Method field becomes active.
3. Select the type of authentication method for the incoming or outgoing back channel, or both.

   Click Help for the field descriptions.

   - If you select the client certificate authentication scheme, add a private key/certificate pair to the certificate data store. The private key/certificate pair is issued from a Certificate Authority.

     Important! The CN of the Subject in the certificate must be the same as the partnership name in the producer to consumer partnership that is configured at the producer.

     For instructions on adding a certificate, see the Policy Server Configuration Guide. Skip this step if the key/certificate pair is already in the data store.
   - If you select No Auth as the authentication method, no additional steps are required.
4. Depending on the authentication method you select, several additional fields are displayed for you to configure.

After entering values for all the necessary fields, the back channel configuration is complete. You can enable SSL on each side of the connection for added security.

Sign and Encrypt Federation Messages

Securing an assertion and encrypting data within the assertion is a critical part of partnership configuration. The Signature step (SAML 1.1) and the Signature and Encryption step (SAML 2.0) let you configure signing and encryption of assertions.

For SAML 2.0, you have the option of choosing a signing algorithm for signing tasks. The ability to select an algorithm supports the following use cases:

- An IdP-->SP partnership in which the IdP signs assertions, responses and SLO-SOAP messages with the RSAwithSHA1, or the RSAwithSHA256 algorithm.
- An SP-->IdP partnership in which the SP signs authentication requests and SLO-SOAP messages with the RSAwithSHA1, or the RSAwithSHA256 algorithm.

Signature verification automatically detects which algorithm is in use on a signed document then verifies it. No configuration for signature verification is required.

Signature Configuration at a SAML 2.0 IdP

The Signature and Encryption step in the partnership wizard lets you define how the product uses private keys and certificates for the following signing functions:

- Sign and verify SAML assertions, assertion responses, and authentication requests.

  For SAML 2.0 POST binding, you are required to sign assertions.

- Sign single logout responses and requests (HTTP-Redirect and SOAP bindings).

There can be multiple private keys and certificates in the certificate data store. If you have multiple federated partners, you can use a different key pair for each partner.

Note: If the system is operating in FIPS_COMPAT or FIPS_MIGRATE mode, all certificate and key entries are available from the pull-down list. If the system is operating in FIPS-Only mode, only FIPS-approved certificate and key entries are available.

To configure signing options

1. Select the Signature and Encryption step in the partnership wizard.
2. In the Signature section, select an alias for the Signing Private Key Alias field. If there is no private key available, click Import to import one. Or, click Generate to create a certificate request.

   By completing this field, you are indicating which private key the asserting party uses to sign assertions, single logout requests and responses.

   Note: click on Help for a description of the fields.
3. Select the hash algorithm for digital signing in the Signing Algorithm field. The IdP signs assertions, responses and SLO-SOAP messages with the specified algorithm.

   Select the algorithm that best suits your application.

   RSAwithSHA256 is more secure than RSAwithSHA1 due to the greater number of bits used in the resulting cryptographic hash value.

   The system uses the algorithm that you select for all signing functions.
4. Select an alias from the certificate data store or the Verification Certificate Alias field.

   By completing this field, you are indicating which certificate verifies signed authentication requests or single logout requests or responses. If there is no certificate in the database, click Import to import one.
5. (Optional) Specify Artifact and POST signature options for the assertion or response or both.
6. (Optional) Specify an SLO SOAP signature option for the logout request, the logout response or both when you are using single logout.

7. (Optional) Select the check box for Require Signed Authentication Requests. This check box verifies that the asserting party only accepts signed requests from the relying party.

   Activate a partnership for all configuration changes to take effect and for the partnership to become available for use. Restarting the services is not sufficient.

If you are using the product in a test environment, you can disable signature processing to simplify testing. Click the Disable Signature Processing check box.

Important! Enable signature processing in a SAML 2.0 production environment.

Encryption Configuration at a SAML 2.0 IdP

The Signature and Encryption step in the Partnership wizard lets you define how the Policy Server uses private keys and certificates to do the following tasks:

- Sign and verify SAML assertions, assertion responses, and authentication requests.

  For SAML 2.0 POST binding, you are required to sign assertions.
- Sign single logout responses and requests (HTTP-Redirect and SOAP bindings).
- Encrypt and decrypt entire assertions, Name IDs and attributes.

There can be multiple private keys and certificates in the certificate data store. If you have multiple federated partners, you can use a different key pair for each partner.

To configure encryption options

1. In the Encryption section, select one or both of the following check boxes to specify the assertion data to be encrypted:
   - Encrypt Name ID
   - Encrypt Assertion
2. Select the certificate alias from the certificate data store for the Encryption Certificate Alias.

   This certificate encrypts assertion data. If no certificate is available, click Import to import one.
3. Select values for the Encryption Block Algorithm and Encryption Key Algorithm fields.

   For the following block/key algorithm combinations, the minimum key size that is required for the certificate is 1024 bits.
   - Encryption Block Algorithm: 3DES

     Encryption Key Algorithm: RSA-OEAP

- Encryption Block Algorithm: AES-256

  Encryption Key Algorithm: RSA-OEAP

  Note: To use the AES-256 bit encryption block algorithm, install Java Cryptography Extension (JCE) Unlimited Strength Jurisdiction Policy Files. You can download these files from http://java.sun.com/javase/downloads/index.jsp.

The encryption configuration is complete.

Partnership Confirmation

Review the partnership configuration before saving it.

Follow these steps:

1. Review the settings in the Confirm step of the Partnership wizard.
2. Click Modify in each group box to change any settings.
3. Click Finish when you are satisfied with the configuration.

The partnership configuration is complete.

Activate the Partnership

After you create a partnership, activate it before you can use it. You can also deactivate a partnership using the same process.

Important! Deactivate a partnership before you modify it.

To activate or deactivate a partnership

1. Select Federation, Partnership Federation, Partnerships.

   The Partnerships dialog opens.

2. From the Actions menu, select Activate or Deactivate next to the partnership of interest.

   A confirm dialog displays.

   Note: Activate is only available for a partnership in DEFINED or INACTIVE status. Deactivate is only available for a partnership in ACTIVE status.

3. Click Yes to confirm your selection.

The status of the partnership is set and the display is refreshed.

Proceed with the Authentication Scheme Setup

After you configure the federated partnerships, the final task at the Administrative UI is to configure and apply an authentication scheme to the redirect JSP resource. This resource invokes the necessary authentication scheme and redirects the user to the target application.

Note: Authentication schemes are only required if the application is protected by OpenID, OAuth, or one of the advanced authentication schemes.

After you apply the authentication scheme, the remaining SSO tasks are done from the User Console, including:

- Configuring an authentication method.
- Configuring an application.

Configure and Apply an OpenID Authentication Scheme

OpenID is an authentication scheme that lets you use an existing account to sign in to multiple web sites, without needing a new password. Users can create accounts with a single OpenID identity provider, and then use those accounts to log on to any website which accepts OpenID authentication.

The SSO service supports OpenID so users can sign on with OpenID providers, such Google and Facebook. The OpenID provider authenticates the user and sends an authentication response. The hosting system verifies the authentication response and completes the authentication process.

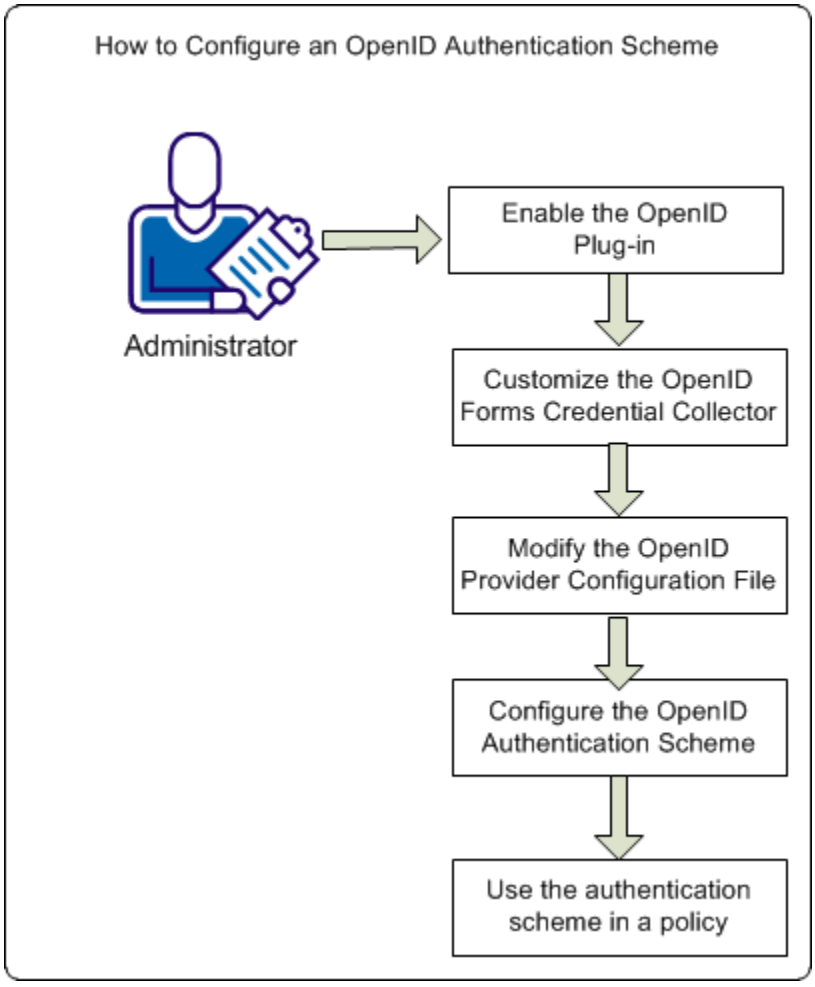
After you configure an OpenID authentication scheme, it can be associated with an OpenID authentication method configured in the User Console.

Prerequisites for the Authentication Scheme

The authentication scheme setup takes place in the Administrative UI. Note the following requirements before configuring the authentication scheme.

- CSP console has been deployed.
- Tenant has been created in the Administrative UI.

This section describes how an administrator can configure the OpenID authentication scheme. The following diagram illustrates the required tasks:



Follow these steps:

1. Enable the OpenID plug–in.
2. Customize the OpenID forms credential collector.
3. Modify the OpenID providers configuration file (see page 117).
4. Configure the OpenID authentication scheme (see page 120).
5. Use the authentication scheme in a policy (see page 120).

Enable the OpenID Plug–in

The OpenID plug-in is referenced in the SPS defaultagent configuration file (webagent.conf). The plug-in is required to let agents communicate with an OpenID provider and communicate the OpenID provider authentication response to the Policy Server.

Contact agent owners and instruct them to enable the required plug-in.

Follow these steps:

1. Log in to the cloud host system.
2. Open the SPS defaultagent configuration file in the following directory:

   <sps home>/proxy-engine/conf/defaultagent

   Where *<sps home>* specifies the SPS installation path.
3. Uncomment line that loads the OpenID plug–in.

   LoadPlugin = *path_to_OpenIDPlugin*

   Example:

   #LoadPlugin="/opt/CA/secure-proxy/agentframework/bin/OpenIDPlugin.so
4. Save the file.
5. Restart the SPS.

Customize the OpenID Forms Credential Collector

A sample OpenID FCC is included with the product. The FCC is required to let users authenticate by:

- Entering an OpenID provider user name.
- Entering a complete OpenID identifier.

By default, the FCC presents numerous OpenID providers. Contact the administrator and instruct them to display only those providers that the protected application supports by modifying the FCC.

Follow these steps:

1. Log in to the cloud host system.
2. Go to the following location:

   <*sps home*>/proxy-engine/examples/siteminderagent/forms

   Where *<sps_home>* specifies the SPS installation path. SPS is part of the CA CloudMinder product.

3. Open the following file with a text editor:

   openid.fcc

4. Review the FCC. Determine if the OpenID providers you require are available or if you have to add a profile. The default providers are located in the following sections:

   var providers_large
   var providers_small

5. Remove the unnecessary providers from the FCC by commenting them. Begin the comment at the provider name. End the comment at the end of the profile.

   Example:

```
/*google : {
      name : 'Google',
      url : 'https://www.google.com/accounts/o8/id'
},*/
```

6. If you have to add a provider, locate the custom provider ID in either the large or small provider sections:

   Example:

```
}/*,
myprovider : {
      name : 'MyProvider',
      label : 'Enter your provider username',
      url : 'http://ca.com/{username}/',
      image : 'images/image.png'
}*/
```

   Note: The separate provider sections correspond to the sizes of provider icons that the FCC displays:

   - The supported size of an icon in the large section is 100 pixels by 60 pixels. The FCC can display up to five large icons.
   - The supported size of an icon in the small section is 24 pixels by 24 pixels. The FCC can display up to 11 small icons.

   a. Add the new provider by removing the following characters:

      /*
      */

b. Update the label and name values. The label value determines the text that the user sees after clicking the provider icon.

Example:

```
myprovider : {
name : 'Foward Inc',
label : 'Enter your Forward Inc user name',
url : 'http://ca.com/{username}/',
image : 'images/image.png'
}
```

Note: Forward, Inc. is a fictitious company name that is used strictly for instructional purposes only and is not meant to reference an existing company.

c. Update the URL value. The URL value represents the OpenID user identifier. The Policy Server forwards the user identifier to the OpenID provider.

Example:

```
myprovider : {
name : 'Foward Inc',
label : 'Enter your Forward Inc user name',
url : 'http://{username}.forwardinc.com/'
image : 'images/image.png'
}
```

d. Update the image value. The image value represents the location of the provider icon that the FCC is to display.

Example:

```
myprovider : {
name : 'Foward Inc',
label : 'Enter your Forward Inc user name',
url : 'http://{username}.forwardinc.com/'
image : 'images/forwardinc.png'
}
```

7. By default, the FCC displays the provider icons in the order in which the provider ID is configured and enabled in the FCC. If you want to change the icon order, adjust the order of the provider IDs accordingly.

   Important! The default provider IDs include the following image index property:

   imageidx

   Do *not* remove or change the property. The property verifies that the FCC displays the correct provider icon.

8. Save the script.

9. Restart the web server.

Modify the OpenID Provider Configuration File

The product provides an OpenID provider configuration file. The file must reference the configuration details of each provider that the protected application supports. If the file does not include the correct settings, authentication fails.

- By default, the file includes sample settings for all of the providers that the OpenID FCC makes available. Review the sample settings and modify them as required.

  Important! The values are samples only. We recommend that you verify all configuration settings with your OpenID provider before deploying the authentication scheme.

- If you added a provider to the FCC, add configuration settings for the provider.

Follow these steps:

1. Log in to the Policy Server host system.
2. Go to the following location:

   siteminder_home\config\properties

   siteminder_home

   Specifies the Policy Server installation path.
3. Do one of the following steps:
   - Open the default provider configuration file:

     Openidproviders.xml

   - Create another instance by copying the default configuration file. Each OpenID authentication scheme that you configure can use its own provider configuration file.

     Example: You can enable Federal Identity, Credential, and Access Management (ICAM) compliance for one instance of the authentication scheme and can disable ICAM compliance for another.
4. Review the file and determine if the OpenID provider settings you require are available or if you have to add settings.
5. If you have to add settings, complete the following steps:
   a. Copy an existing OpenID provider node and all of its child nodes. All required and optional nodes are included within the following nodes:

      <OpenIDProvider>
      </OpenIDProvider>

   b. Add the new OpenID provider node and all of its child nodes to the following root node:

      <TrustedOpenIDProviders>
      </TrustedOpenIDProviders>

6. Configure the settings for each provider that the authentication scheme is to support using the following node descriptions:

OpenIDProvider *RequestType="value"*

Indicates the beginning of the configuration settings for a provider.

RequestType

(Optional) Specifies the schema type that the provider supports.

Valid values: ax or sreg.

Default: ax.

ProviderName

Specifies the URL of the OpenID provider hosting the service. The value can include a comma–separated list of provider URLs.

Required Claims

Specifies the claims that the OpenID provider returns as part of the authentication request. If the provider cannot provide all of the required claims, authentication fails. This node requires at least one claim node.

Claim

Defines an individual required claim.

URI

Specifies the URI form of the OpenID provider claim. The Policy Server constructs the authentication request using this value.

Important! Verify that the value of the first required claim maps to a user attribute in your user directories. The Policy Server determines the value of the first required claim that is based on the provider authentication response. The Policy Server then searches all user directories in the policy domain for a user that matches the claim value. If the Policy Server cannot map the claim value to a user attribute, authentication fails.

Value: The value must adhere to the type of schema that the provider supports.

Alias

(Optional) Defines the user-friendly name of the URI node value and prevents the URI from being stored or referenced. The system uses the alias to identify the claim.

Value: Any string.

Example: Instead of storing a URI that returns the first name of users in the session store, the system can reference the claim name as fullname.

Note: The system appends the following prefix to an alias that is stored in the session store:

smopenidclaim

Optional Claims

(Optional) Specifies the optional claims that the OpenID provider is to return as part of the authentication request. If the provider cannot provide an optional claim, authentication does not fail. This node requires at least one claim node.

Pape

(Optional) Defines the properties that ICAM compliance requires. If you are configuring the authentication scheme for ICAM compliance, this node and all child nodes are required.

max_auth_age

(Optional) Specifies the time for which the OpenID provider user session is valid. If the user session is valid, the OpenID provider authenticates the user for a protected resource using a provider–specific cookie. If the session expires, the user is prompted to reauthenticate.

Unit of measurement: seconds.

Default: 0.

If you leave the default value, the user must authenticate against the OpenID provider, regardless of a valid session.

Value: The value must be a positive integer.

Policies

(Optional) Specifies a comma–separated list of the ICAM policies to which the OpenID provider must adhere. If the provider does not adhere to the compliance level, authentication fails.

7. Save and close the file.

Configure an OpenID Authentication Scheme

Configure an OpenID authentication scheme when using an external IdP to authenticate users for SSO application requests.

Follow these steps:

1. Click Infrastructure, Authentication.
2. Click Authentication Schemes.
3. Click Create Authentication Scheme.

   Verify that the Create a new object of type Authentication Scheme is selected.

   Click OK
4. Enter a name for the scheme that indicates its purpose.
5. Specify a protection level.
6. Select OpenID Template from the Authentication Scheme Type list.

   Scheme-specific fields and controls appear.
7. Complete the fields:

   Use Relative Target

   Select the check box. Disregard the values for Web Server Name/Port.

   Target

   /siteminderagent/forms/openid.fcc

   This is the default string.
8. (Optional) Select Persist Authentication Session Variables to store user data in the session store.

   If you are not using the session store, set the following fields:

   Pre Processing Chain

   com.ca.sm.openid.command.StoreClaimsToContext
9. Disregard the remaining fields and click Submit.

   The authentication scheme is saved and can be assigned to a realm.

Use the Authentication Scheme in a Policy

For SSO applications, you must establish a one-to-one correspondence between an authentication method configured at the User Console and an authentication scheme configured at the Administrative UI. The authentication method and the scheme work together to enforce user authentication for a requested SSO application.

After you create an authentication scheme, the scheme has to protect the authentication URL specified for a given authentication method. To protect the URL, the scheme is assigned to a realm, and the realm becomes part of a policy.

Follow these steps::

1. Configure the policy domain for the tenant.
2. Assign user directories to the tenant domain.
3. Create a realm and rule for the tenant domain.
4. Create a policy to protect the authentication URL.

Select the Policy Domain for the Tenant

A policy domain protects logical groupings of resources. When you deploy a tenant in the Administrative UI, the system automatically generates a policy domain for the tenant.

Work with this domain when establishing a policy for the authentication URL.

Follow these steps:

1. In the Administrative UI, click Policies, then click Domain.
2. Click Domains.

   The Domains page appears.
3. Select the domain for the appropriate tenant and modify it.
4. Confirm that the tenant user directory is part of the tenant domain. If it is not, add the user directory to the domain (see page 121).
5. Click Submit.

   The policy domain is modified.

Assign User Directories to the Tenant Domain

You can add one or more tenant user directories to a tenant domain. The system authenticates users by comparing the user credentials to the credentials that are stored in the user directories. The system searches the user directories in the same order that they are listed in the policy domain.

Note: The following procedure assumes that the tenant user directory is already configured in the Administrative UI.

Follow these steps:

1. Click Policies, Domain.
2. Click Domains.

   The Domains page appears.
3. Specify the search criteria and click Search.

   A list of domains that match the search criteria appears.
4. Click the name of the domain that you want to modify.

   The View Domain page appears.
5. Click Modify.

   The settings and controls become active.
6. In the General tab, click Add/Remove.

   The Choose user directories page appears.
7. Select one or more user directories from the list of Available Members, and click the right-facing arrows.

   The user directories are removed from the list of Available Members and added to the list of Selected Members.

   Note: To select more than one member at one time, hold down the Ctrl key while you click the additional members. To select a block of members, click the first member then hold down the Shift key while you click the last member in the block.
8. Click OK.

   The selected user directories are listed under User Directories.

   Note: To create a user directory and add it to the domain, click Create.
9. Click Submit.

   The selected user directories are added to the domain.

Configure a Realm and a Rule for the Tenant Domain

A realm groups resources that have similar security requirements and share a common authentication scheme. For the tenant domain, create a realm and associate it with a Web Agent.

Note: The following procedure assumes that you are creating an object. You can also copy the properties of an existing object to create an object.

Follow these steps:

1. Click Policies, Domain, Realms.

   The Realms page appears.

2. Click Create Realm.

3. Select the tenant domain that you want to modify, and click Next.

4. Type the name and a description of the realm.

   Specify a name that indicates the realm is for an SSO authentication URL. For example:

   - OAuth_Google_Realm
   - OAuth_Facebook_Realm

5. Click Lookup Agent/Agent Group to select an agent.

6. Select the **cam-agent** and click OK.

7. Specify the Resource Filter for the authentication scheme you are using. This scheme has to tie in to the authentication method chosen in the User Console configured and applied to the application.

   The following list includes the resource filter for all available authentication schemes for cloud SSO. Use the resource filter for your authentication scheme.

   For HTML Forms authentication

   For environments created in CA CloudMinder 1.51 or later:

   /chs/redirect/*tenant_tag*/forms

   For environments created before CA CloudMinder 1.51:

   /affwebservices/<tenant-name>/forms.jsp

   For OpenID authentication

   /affwebservices/*tenant_tag*/*duplicate_openid_file*.jsp

   Copy the default openid.jsp file to a unique name, such as openid-google.jsp. Having a unique jsp file is necessary to distinguish openID configurations.

   For OAuth authentication

   /affwebservices/*tenant_tag*/*duplicate_oauth_file*.jsp

   Copy the default oauth.jsp file and give the copy a unique name, such as oauth-google.jsp or oauth-facebook.jsp. Having a unique jsp file is necessary to distinguish OAuth configurations.

For Arcot PKI authentication scheme

For environments created in CA CloudMinder 1.51 or later:

/chs/redirect/*tenant_tag*/arcotid

For environments created before CA CloudMinder 1.51:

/affwebservices/<tenant-name>/arcotid.jsp

For Arcot OTP authentication scheme

For environments created in CA CloudMinder 1.51 or later:

/chs/redirect/*tenant_tag*/arcototp

For environments created before CA CloudMinder 1.51:

/affwebservices/<tenant-name>/arcototp.jsp

For Arcot PKI Risk authentication scheme

For environments created in CA CloudMinder 1.51 or later:

/chs/redirect/*tenant_tag*/arcotidrisk

For environments created before CA CloudMinder 1.51:

/affwebservices/<tenant-name>/arcotidrisk.jsp

For Arcot OTP Risk authentication scheme

/chs/redirect/*tenant_tag*/arcototp_risk

For environments created in CA CloudMinder 1.51 or later:

/chs/redirect/*tenant_tag*/arcototprisk

For environments created before CA CloudMinder 1.51:

/affwebservices/<tenant-name>/arcototprisk.jsp

tenant_tag is a unique identifier for a tenant. You specify the tag when deploying a tenant environment in the Administrative UI. To view a list of tags, select the Tenants tab.

8. Complete the remaining fields:

Default Resource Protection

Protected

Authentication Scheme

Select the authentication scheme that you configured for the SSO application and the scheme that corresponds to the resource filter. For example, if you are using OpenID and you configured a scheme named OpenID Auth, select that scheme.

9. Create a rule:
   a. Specify a name for the rule.

      For example, if Google is the OAuth provider, name the rule oauth_googlerule. If Facebook is the OAuth provider, name the rule oauth_facebookrule.

   b. In the Realm and Resource area, edit the Resource value by deleting the forward slash (/) character.

      Important! The Resource value is now the asterisk (*) character only.

   c. In the Action area, select Web Agent actions.

   d. Under Action, control-click to multi-select GET, HEAD, and POST.

   e. Accept the defaults for the remaining settings.

   f. Click Ok.

10. Specify the session properties.

    Note: Click Help for a description of fields, controls, and their respective requirements.

11. Skip the other configuration options.

12. Click Finish.

    The realm is complete.

Create the Policy to Protect the Authentication URL

Create a policy for the domain. Policies define relationships between users and resources. The policy components work together and protect the resource.

After you create the policy, you add users and rules.

Follow these steps:

1. Click Policies, Domain.
2. Click Domains.
3. Specify search criteria, and click Search.

   A list of domains that match the search criteria appears.

4. Click the edit icon next to the domain for which you want to create a policy.

   The Modify Domain page appears.

5. Click the Policies tab.

   The Policies page appears.

6. Click Create.

   The Create Policy page appears.

7. Enter a name and a description for the policy. Use a name that indicates that the policy is for the authentication URL.
8. Add individual users, user groups, or both from the Users tab. The users are members of the tenant user directory associated with the domain. When a user tries to access a protected resource, the policy verifies whether the user is allowed to access the resource.

   Note: If you select Add Members, the User/Groups pane opens. Individual users are not displayed automatically. Use the search utility to find a specific user within one of the directories.

   You can edit or delete a user or group by clicking the right arrow (>) or minus sign (-), respectively.
9. When you have finished selecting users, user groups or both, click OK.
10. Add rules to the policy from the Rules tab.

    Rules indicate which resources are part of a policy and whether to allow or deny access to the resources.

    Note: Add at least one rule or rule group to a policy.

    The Available Rules pane opens.
11. Select the rule that you created for the authentication URL resource.

    For example, if you configured a rule specific to Google, named oauth_googlerule, select that rule.

    You are not required to configure a response for the rule.
12. Click OK.
13. Click Submit to save the policy.

The policy configuration is complete.

Create the Authentication Method

An authentication method represents how an application is protected. After you configure an authentication method, you assign it to the application you want to protect. Multiple applications can use the same authentication method. A single application can reference multiple authentication methods.

Configure an authentication method that satisfies the protection requirements for an application.

Note: The system creates authentication methods corresponding to each of the advanced authentication flows. If you are configuring Advanced Authentication for the tenant, do not create an authentication method. Modify the existing authentication method as described in this procedure.

Follow these steps:

1. Log in to the User Console.
2. Navigate to Applications, Authentication Methods, Create an Authentication method.
3. In the top section of the Create Authentication method screen, complete the following fields:

   Name

   Enter a string that identifies the authentication method you are configuring.

   Description

   Enter a description for the authentication method. The login page displays this description as a label.

   Enabled

   Select this check box to make the authentication method immediately available.

4. In the Configure Authentication Method section, select one of the following options and enter the authentication URL for that option.

   When the authentication method is associated with an application, the authentication service appends the redirect URL for the application.

   Note the following variables in the URLs:

   cloud_host is the CA CloudMinder system.

   local_entity_ID is the name of the local entity that is specified in the IdP-to-SP partnership, which is configured at the CSP console.

   remote_entity_ID, consumer_entity_ID or resource_partner_ID is the name of the remote entity that is specified in the configuration of the asserting-to-relying party partnership. The partnership is configured at the CSP console.

Basic

Represents a form-based authentication scheme that uses the basic credentials of a user name and a password. The basic authentication method corresponds to the HTML Forms authentication scheme in the Administrative UI.

Enter the authentication URL of the following format:

http://*cloud_host*:*port*/chs/redirectservlet/*tenant_tag*/forms

tenant_tag is a unique identifier for a tenant. You specify the tag when deploying a tenant environment in the Administrative UI. To view a list of tags, select the Tenants tab.

External IDP—Google or Facebook

Represents a third-party identity provider (IdP) that authenticates users. Social media sites, such as Google or Facebook can serve as external IdPs. Other federated partners that support the SAML and WS-Federation protocols can also serve as external IdPs.

If Google or Facebook is acting as the third-party IdP, specify the OpenID or OAuth authentication method. Each site supports both protocols.

Enter the relevant URL for the protocol, as shown:

OpenID

http://*cloud_host*:*port*/affwebservices/*tenant_tag*/*duplicate_openid_file*.jsp

When configuring the OpenID authentication scheme at the Administrative UI, the default openid.jsp file is copied and given a unique name, such as openid-google.jsp. Having a unique jsp file is necessary to distinguish OpenID configurations.

The default JSP file is located in the directory /opt/CA/secure-proxy/Tomcat/webapps/affwebservices/redirectjsp.

OAuth

http://*cloud_host*:*port/*affwebservices/*tenant_tag*/*duplicate_oauth_file*.jsp

When configuring the OAuth authentication scheme in the Administrative UI, the default oauth.jsp file is copied and given a unique name, such as oauth-google.jsp. Having a unique jsp file is necessary to distinguish OAuth configurations.

The default JSP file is located in the directory /opt/CA/secure-proxy/Tomcat/webapps/affwebservices/redirectjsp.

tenant_tag is a unique identifier for a tenant. You specify the tag when deploying a tenant environment in the Administrative UI. To view a list of tags, select the Tenants tab.

External IDP—Other

Select Other when a SAML or WS-Federation-compliant partner is the IdP. The federation profiles SAML 1.1, SAML 2.0, and WS-Federation 1.2 are all supported.

Enter the relevant URL for the protocol, as shown.

For SAML 1.1 transactions
http://*cloud_host.domain:port*/affwebservices/public/intersitetransfer?CONSUMERID=*consumer_entity_ID*&TARGET=http://*consumer_site/target_url*

For SAML 2.0 SP-initiated transactions

http://*cloud_host.domain:port/*affwebservices/public/saml2authnrequest?ProviderID=*local_entity_ID*&RelayState=http://*sp_site/target_url*

For SAML 2.0 IdP-initiated transactions

http://*cloud_host.domain:port/*affwebservices/public/saml2authnrequest?SPID=*remote_entity_ID*&RelayState=http://sp_site/*target_url*

For WS-Federation IP-initiated transaction

http://*cloud_host.domain:port*/affwebservices/public/wsfeddispatcher?wa=wsignin1.0&wtrealm=*resource_partner_ID*&wctx=*target_url*

Advanced Authentication

Represents one of the authentication protocols that the CA CloudMinder Advanced Authentication Service provides.

Select one of the following options and the URL is entered automatically:

For ArcotID PKI Only

For environments created in CA CloudMinder 1.51 or later:

https://cl*oud_host*:p*ort*/chs/redirectservlet/*tenant_tag*/arcotid

For environments created before CA CloudMinder 1.51:

https://cl*oud_host*:p*ort*/affwebservices/<tenant-name>/arcotid.jsp

For ArcotID PKI with Risk

For environments created in CA CloudMinder 1.51 or later:

https://cl*oud_host*:p*ort*/chs/redirectservlet/*tenant_tag*/arcotidrisk

For environments created before CA CloudMinder 1.51:

https://cl*oud_host*:p*ort*/affwebservices/<tenant-name>/arcotidrisk.jsp

For ArcotID OTP Only

For environments created in CA CloudMinder 1.51 or later:

https://cl*oud_host*:p*ort*/chs/redirectservlet/*tenant_tag*/arcototp

For environments created before CA CloudMinder 1.51:

https://cl*oud_host*:p*ort*/affwebservices/<tenant-name>/arcototp.jsp

For ArcotID OTP with Risk

For environments created in CA CloudMinder 1.51 or later:

https://cl*oud_host*:p*ort*/chs/redirectservlet/*tenant_tag*/arcototprisk

For environments created before CA CloudMinder 1.51:

https://cl*oud_host*:p*ort*/affwebservices/<tenant-name>/arcototprisk.jsp

tenant_tag is a unique identifier for a tenant. You specify the tag when deploying a tenant environment in the Administrative UI. To view a list of tags, select the Tenants tab.

5. Click Submit.

The authentication method is available to protect an application.

Create an Application

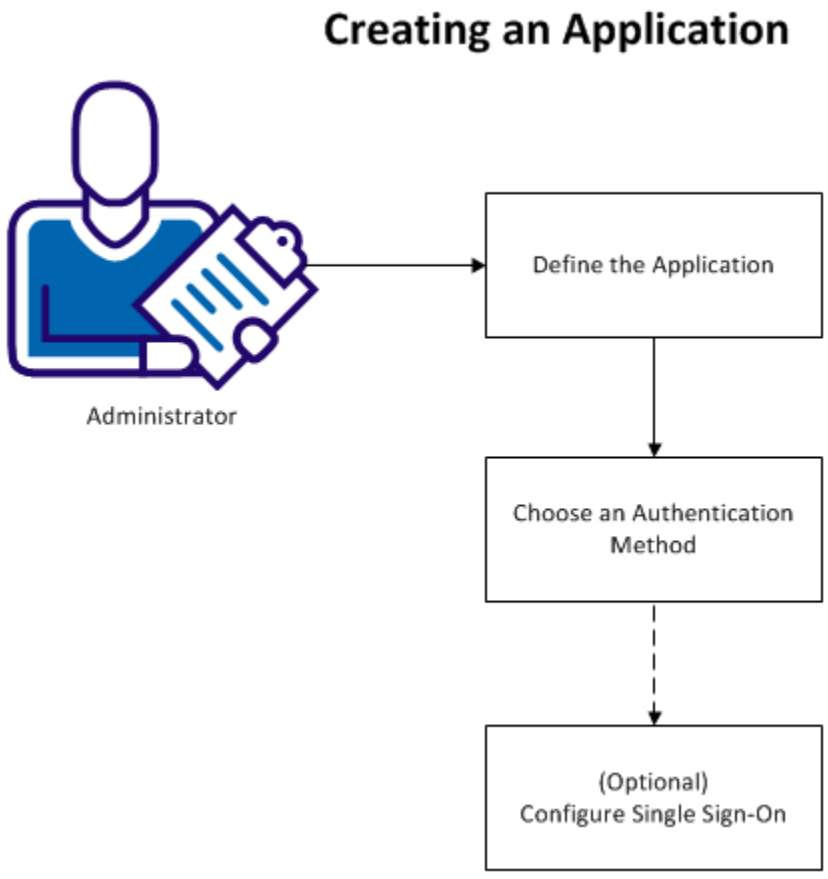
As an administrator, you want to give your users secure and convenient access to software resources. For example, your users need access to your email system, which can be hosted on-premise by your organization. Users also need access to Salesforce.com, which an external organization hosts in the cloud.

Note: Although a tenant administrator typically creates applications for their environment, a hosting administrator can also perform this task.

You create an application to define how users access a software resource. For example, when you configure an application, you define what type and level of security protects the resource. If you have purchased CA CloudMinder Advanced Authentication, you can configure advanced security such as two-factor authentication to protect the resource. If you have purchased the CA CloudMinder Single Sign-on service, you can configure SSO for the application. Users only log in once to access all applications that are configured for SSO.

Once an application is configured, you can give users access to the software resource. You can configure a service that includes the application, and can assign the service to users. The users can click the icon in the User Console Home Page to access the application. For more information, see Creating a Service Using the Service Wizard (see page 64). You can also give users access to the application by making a link to the application available. For example, you can insert the link into any web page or you can send the link in an email.

The following diagram shows the information to understand and the steps to perform to create an application and make it available to users.



The following topics explain how to create an application:

1. Define the Application Profile (see page 59)
2. Choose an Authentication Method (see page 61)
3. (Optional) Configure Single Sign-On (see page 63)

Define the Application

You define the application details through the User Console.

Follow these steps:

1. Log in with an account that has application management privileges.

   For example, the default Tenant Administrator role has the appropriate privileges.

2. From the navigation menu, select Applications.

3. Click Applications, then Create Application.

   The Create Application screen appears.

4. Enter a name and description.

5. Associate a group with the application, if desired.

   Only the users who are members of the indicated group receive access.

   Note: If you are configuring the application for SSO access, the group that you choose must match the group name that is indicated in the SSO partnership configuration for this application. Only if the group names match will the system restrict access to group members. To confirm the group name that is indicated in the partnership configuration, refer to your hosting administrator. SSO partnership configuration information is available in the CSP Console.

6. Enter a launch URL for the Application.

   A launch URL is the fully qualified domain name of the software resource you want to make available to users. For example, if a user clicks the icon for this application in the User Console Home page, they are directed to the launch URL.

   If you are configuring the application for SSO access, the launch URL is the SSO Service URL generated during SSO partnership configuration. Refer to your hosting administrator for this information.

   If you are not configuring an SSO application, simply enter the fully qualified domain name of the software resource. Use the following format:

   https://softwareresourcedomainname.com

7. Choose a logo.

   This logo is the icon for the application that appears in the User Console Home page. Users can click the icon to access the software resource.

   Note: You can also give users access to the application by inserting a link to the application into any web page.

8. Enter a welcome message.

   When users click any link you provide to the application, a login screen appears. The welcome message appears at the top of this screen.

9. Select a self-registration task.

   If a user attempts to access the application but the user does not have a CA CloudMinder account, you can allow them to self-register. Choose one of the following self-registration tasks:

   Create New Account

   Presents a simple registration form. Upon submission, creates a user account.

   Create New Account with Workflow

   Presents a simple registration form. Upon submission, forwards the user account request to one or more approvers. Creates an account upon approval.

   Create New Account with Domain Validation

   Presents a simple registration form. Upon submission, compares the email domain of the user to the tenant email domain. If they match, sends a confirmation email to the user. Creates an account upon user confirmation.

   Note: The tenant email domain is specified in the User Console, under Tenant Administration, Tenant Settings.

   Self-Registration with Attribute Exchange

   Do not choose this self-registration task in the context of application access. This task is intended for a separate purpose.

10. Choose an authentication method. (see page 61)

Choose an Authentication Method

In the Create Application screen, continue the process of creating an application by choosing one or more authentication methods. When a user attempts to access the application, the system presents a login screen. The authentication methods that you choose appear on this screen. The user can log in using their choice of the available authentication methods.

For example, you can select the Basic and Google External IDP authentication methods for an application. The application login screen displays user name and password fields for basic authentication. The login screen also displays the Google icon, so users can log in with their Google credentials.

Follow these steps:

1. In the Authentication Methods area, click Add.

   The Select Authentication Methods screen displays a list of the authentication methods available in the tenant environment.

   Note: First, create authentication methods in the system before you perform this step. You define authentication methods through the User Console, using the Authentication Methods tasks. For more information, see Create Authentication Methods.

2. Select one or more authentication methods. The following types of authentication method are available:

   Basic

   Offers simple user name and password login.

   External IDP

   Offers log in through an external credential provider, such as Google or Facebook.

   Advanced Authentication

   Offers advanced authentication methods that have been configured for your environment, such as One Time Password (OTP) authentication.

   Note: Advanced Authentication methods only appear if you have purchased the Advanced Authentication Service.

   You can choose as many authentication methods, of any type, as you want. All the methods that you select are displayed on the login page that appears when a user attempts to access the application.

3. Click Select.

   The Create Application screen appears, updated with the list of authentication methods you selected.

4. (Optional) From the drop-down list, choose a default authentication method.

   Note: Advanced Authentication methods are never available as a default.

5. Configure Single Sign-On (see page 63).

(Optional) Configure Single Sign-On

Note: The option to configure single sign-on settings only appears in the User Console if you have purchased the SSO service.

During partnership configuration for an SSO application, a hosting administrator specifies a *federation attribute* for the partnership. The system uses this attribute to exchange information with the target software resource during single sign-on operations. For example, when configuring an SSO partnership between CA CloudMinder and salesforce.com, a hosting administrator chooses User ID as the federation attribute. The system retrieves this attribute from the database and forwards it in a SAML assertion to salesforce.com to facilitate single sign-on.

Some target software resources require the federation attribute to have a specific format. If this format differs from the format CA CloudMinder uses for the attribute, use the following steps to set the attribute value to the required format. This process is named setting the rule string for the attribute.

Note: Only configure the rule string if the software resource requires that the attribute take a format different from the way it is stored in the CA CloudMinder database.

Follow these steps:

1. In the Create Application screen, click Configure Single Sign On settings for the application.

   The Single Sign On configuration settings appear.

2. Select the Federation User Attribute.

   The attribute that you choose must match the assertion attribute that is indicated in the SSO partnership configuration for this application. If the attribute names do not match, users cannot successfully access this application through SSO. To confirm the assertion attribute name that is indicated in the partnership configuration, refer to your hosting administrator. SSO partnership configuration information is available in the CSP Console.

3. Configure the rule string for the Federation User Attribute.

   The rule string is the format that you want the attribute to take when the system passes it to the target software resource.

   Note: To learn the exact format that is required for this attribute, refer to your hosting administrator, or an administrator at the target software resource.

You have created an application and applied an authentication method. You have also configured single sign-on settings if applicable. You can now include this application in a service so that users can access the application.

Make a Software Resource Available to Users

As an administrator, you want to make a software resource that you configured through CA CloudMinder available to users. Depending on the service you purchased, you can configure this software resource for one or more of the following services:

- Single sign-on (SSO)
- Advanced authentication
- Provisioning
- Self-registration

Before you make the resource available to users, confirm that you have performed the following steps, if applicable. (The responsible administrator is indicated in parentheses.)

- For SSO, configure one or more partnerships (hosting administrator)
- Configure and apply an authentication scheme (hosting administrator)
- Create authentication methods (tenant administrator)
- For automatic account creation, create roles provisioning roles (tenant administrator)
- Create an application, optionally enabling self-registration (tenant administrator)
- Create a service (tenant administrator)

When you have completed all configuration, you can make the software resource available to users. You can make the resource available by using one or more of the following methods:

1. Assign the service to a user (see page 70).
2. Allow users to request access to the service.

   In the CA CloudMinder User Console, when the user clicks My Access, then Request & View Access, the user sees a list of services available for their request. The services that appear in this list are those marked "Self Subscribing" service creation.

   When the user requests access, the system assigns the service to the user. The user receives all applications, roles, groups, and attributes that are associated with the service. If the service includes a Launch Role for an application, an icon and a link to the application appear in the User Console Home page.

3. Give users a link to the software resource.

   Give users access to the application by making a link to the application available. For example, you can insert the link into any web page or you can send the link in an email.

   You can use the link from the Administrative UI. From the Federations tab, select Partnership Federation, Partnerships. Select the appropriate partnership from the list to display the partnership configuration. In the SSO section, locate the link labeled *SSO Service URL.* To provide access to the application, use this link.

Assign a Service to a User

You can assign a service directly to an individual user. This user becomes a *member* of the service.

Follow these steps:

1. Navigate to Services, Request & View Access.

   A list of services you can administer appears.

2. Select the service that you want to assign to a user and click Select.

   A list of users that are assigned to the service appears.

3. Click Request Access.

4. Search for a user to whom you want to assign the service.

   To display a list of all users for whom you have administrative privileges, click Search without modifying the search criteria.

5. Select a user and click Select.

   An updated list of users that are assigned to the service appears.

6. Click Save Changes.

   The user receives the specified service. The user receives all applications, roles, groups, and attributes you included in the service.

Chapter 4: Configure and Apply an OAuth Authentication Scheme

OAuth is an open standard for authorization. It enables users to share resources without sharing their identity. You can permit access to one site for resources stored on another site.

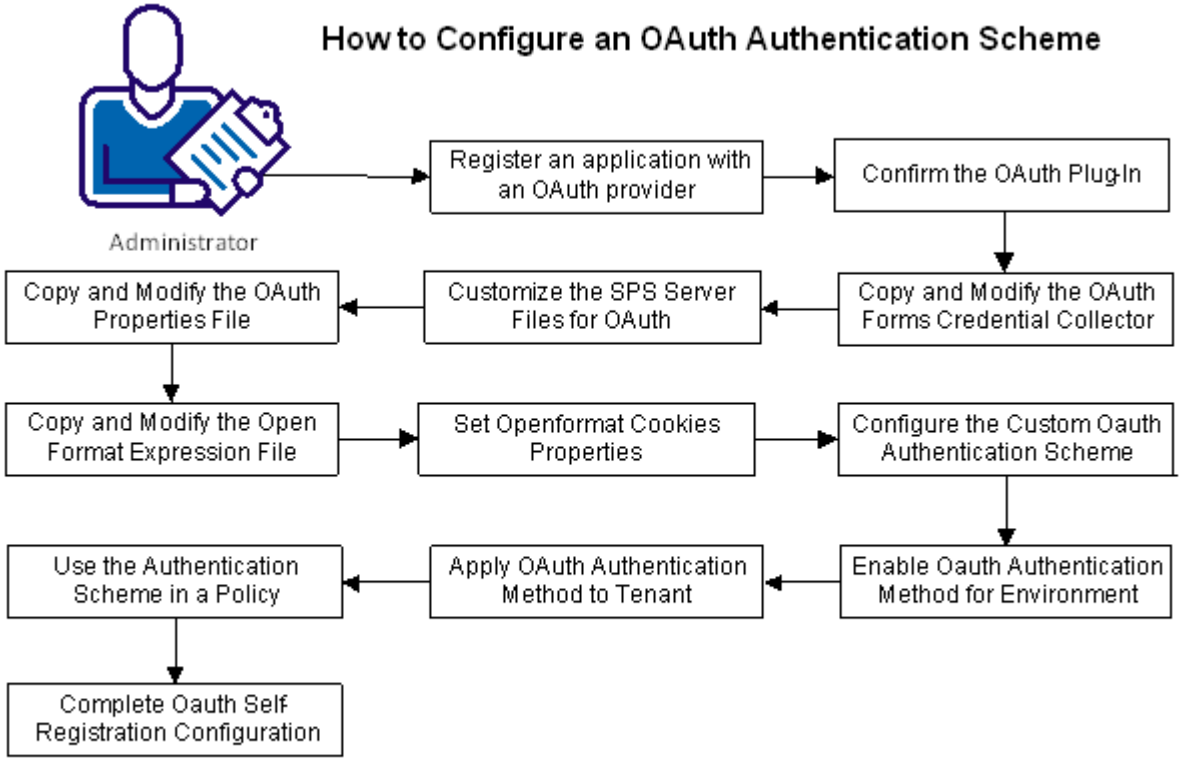
The host Policy Server performs its authentication by first directing the user to the authorization server. User authentication is initially done by an OAuth authorization server, which issues an authorization token upon a successful authentication. The token is the means by which the Policy Server retrieves user information then authenticates the user based on these claims. Upon successful authentication, the user gets access to the target resource.

Users can sign-on with OAuth authentication servers, such as Google and Facebook. Users can authenticate themselves with OAuth credentials and access a protected resource.

If a user does not have a CA CloudMinder account, you can enable self-registration. The first time a user attempts to sign-on through an OAuth provider, the user is prompted to create a CA CloudMinder account. The user can then proceed with sign-on. On subsequent log-ins, the user signs on through the OAuth provider with no interruption.

These instructions provide the steps to configure OAuth authentication between CA CloudMinder as the Service Provider and Google or Facebook as the Identity Provider. These instructions also provide the steps to configure self-registration for Oauth authentication.

The following figure shows the configuration procedure. Before you begin OAuth configuration, we strongly recommend that you review this entire OAuth configuration process document.



Complete the configuration tasks:

1. Register an application with an OAuth provider. (see page 140)
2. Confirm the OAuth Plug-In (see page 142).
3. Copy and Modify the OAuth Provider Configuration File (see page 143).
4. Customize the SPS Server Files for OAuth. (see page 147)
5. Copy and Modify the OAuth Properties File. (see page 148)
6. Copy and Modify the OAuth Open Format Expression File (optional) (see page 150).
7. Set Openformat Cookies Properties (optional). (see page 152)
8. Configure the Custom OAuth Authentication Scheme (see page 153).
9. Enable OAuth Authentication Method for Tenant Environment. (see page 155)
10. Apply OAuth Authentication Method to Tenant. (see page 155)
11. Use the Authentication Scheme in a Policy (see page 120).
    a. Select the Policy Domain for the Tenant. (see page 121)
    b. Assign User Directories to the Tenant Domain (see page 121).
    c. Configure a Realm and a Rule for the Tenant Domain (see page 122).
    d. Create the Policy to Protect the Authentication URL (see page 125).
12. Complete OAuth Self-Registration Configuration. (see page 162)
    a. Create a Rule for Self-Registration. (see page 163)
    b. Create a Response for Self-Registration. (see page 164)
    c. Add Self-Registration Rule and Response to the Policy. (see page 166)

Register an Application with an OAuth Provider

Prerequisites

Create a Google Apps or a Facebook account and register the application.

For Google

Follow these steps:

1. Establish a Google Apps account.
2. Navigate to https://code.google.com/apis/console and log in.

3. Select API Access and create an OAuth client ID. To create the client ID, enter the following information:

   Product name

   Product logo

   Enter the location of your product logo image.

   Home Page URL

   http(s)://homepage.com

   Example: http://www.forwardinc.com

   The preceding branding information is shown to users whenever they request access to your application.

   Application type

   Web application

   Your site or hostname

   http(s)://cloudminder_host

   Example: https://cloud.ca.com

4. Click Create clientID.
5. Click Edit settings and edit the Redirect URI as follows:

   Authorized Redirect URIs

   https://cloudminder_host/siteminderagent/forms/oauthcb.fcc?SMQUERYDATA=Sample

   Example:
   https://cloud.ca.com/siteminderagent/forms/oauthcb.fcc?SMQUERYDATA=Sample

   Important! If the SecureURLs parameter for the CAM-AgentObj object is set to "No", instead edit the Redirect URI as follows:

   http://cloudminder_host/siteminderagent/forms/oauthcb.fcc

   By default, the SecureURLs parameter is set to "Yes" during SiteMinder Policy Server installation. You can check the value of the SecureURLs parameter in the CSP Console. Log in to the CSP Console, then click Infrastructure, Agent, Agent Configuration Objects. Select CAM-AgentObj and click Edit. Page forward to the SecureURLs parameter.

   Authorized JavaScript Origins

   http(s)://cloudminder_host

   Example: https://cloud.ca.com

6. Click Update.

The Google registration process is complete.

For Facebook

Follow these steps:

1. Establish a Facebook account.
2. Go to https://developers.facebook.com/apps.
3. Choose the AppName and click Continue.
4. Enter the captcha text displayed and click Continue.
5. In the Website with Facebook Login section, complete the following field:

   Site URL

   https://cloudminder_host/siteminderagent/forms/oauthcb.fcc?SMQUERYDATA=Sample

   Example:
   https://cloud.ca.com/siteminderagent/forms/oauthcb.fcc?SMQUERYDATA=Sample

   Important! If the SecureURLs parameter for the CAM-AgentObj object is set to "No", instead edit the Redirect URI as follows:

   http://cloudminder_host/siteminderagent/forms/oauthcb.fcc

   By default, the SecureURLs parameter is set to "Yes" during SiteMinder Policy Server installation. You can check the value of the SecureURLs parameter in the CSP Console. Log in to the CSP Console, then click Infrastructure, Agent, Agent Configuration Objects. Select CAM-AgentObj and click Edit. Page forward to the SecureURLs parameter.
6. Click Save changes.

The registration process results in the generation of the client application URL, the client application ID, and its associated secret. Registration also generates the OAuth authorization server endpoint URLs, from where the OAuth service obtains the authorization code and access token. Some of this information is required when setting up the files that the OAuth authentication scheme uses to operate properly.

Confirm the OAuth Plugin

The OAuth plug-in is referenced in the Secure Proxy Server configuration. The plug-in lets agents communicate with an OAuth provider and forward the OAuth response from the OAuth provider to the Policy Server.

Typically, the OAuth plugin loads on installation. Confirm that the OAuth Plugin is loaded.

Follow these steps:

1. Log in to the cloud host system.
2. Navigate to the following directory:

   web_server_home/secure-proxy/proxy-engine/conf/defaultagent

   web_server_home

   Specifies the web server installation path.
3. Open the WebAgent.conf file.
4. Confirm that the following entry exists in the file and is uncommented:

   LoadPlugin = *path_to_OAuthPlugin*

   Example:

   #LoadPlugin="/opt/CA/secure-proxy/agentframework/bin/libOAuthPlugin.so
5. If necessary, add or uncomment the preceding entry.
6. If you made changes, save the file and restart the web server.

Copy and Modify the OAuth Provider Configuration File

An OAuth provider configuration file (oauthproviders.xml) is installed with the Policy Server. The provider configuration file contains configuration details of each provider and the protected application. If the file does not include the correct settings, authentication fails.

Information about the file:

- By default, the file includes sample settings for all of the providers that the OAuth FCC makes available. Review the sample settings and modify them as required.

  Important! The values are samples only. We recommend that you verify all configuration settings with your OAuth provider before deploying the authentication scheme.
- The provider configuration is separated from the registered applications configuration to reuse the provider configuration for multiple applications.
- Providers configuration details must follow application configuration details. Each application uses one of the predefined provider configurations.
- Each application must have a PROVIDERLINK to one provider with which it is registered.
- If you configure multiple OAuth authentication schemes, each scheme can use its own provider configuration file (oauthproviders.xml).

Follow these steps:

1. Log in to the Policy Server host system.
2. Go to the following location:

   siteminder_home/config/properties

   siteminder_home

   Specifies the Policy Server installation path.
3. Copy the oauthproviders.xml file and name the copy to reflect the tenant.

   Note: You do not need to create a separate oauthproviders.xml file per OAuth provider. Settings per provider are indicated within the file. You only need a separate oauthproviders.xml file per tenant.

   Examples:

   - oauthproviders-tenant1.xml
   - oauthproviders-tenant2.xml
4. Open the file copy.
5. Review the file and determine if the necessary OAuth provider settings are available. To add a provider, complete the following steps:

   a. Copy an existing OAuth provider node and all of its child nodes. All provider nodes are included within the following root node:

   ```
   <OAuthProvider>
   </OAuthProvider>
   ```

   b. Add the new OAuth provider node and all of its child nodes under the following root node:

   ```
   <TrustedOAuthProviders>
   </TrustedOAuthProviders>
   ```

6. Configure the settings for each provider. Update values for the following settings:

OAuth providername

Identifies the OAuth provider for this node. Enter the name of the provider.

Note: Use lower-case when entering a provider name.

AuthorizationURL

Provides the authorization server end-point URL for this provider. This URL must generate an authorization token after successful authentication of a user.

Google example: https://accounts.google.com/o/oauth2/auth

Facebook example: https://www.facebook.com/dialog/oauth

AccessTokenURL

Provides an access token end-point URL. A user can query for an access token by exchanging authorization code along with application configuration details.

Google example: https://accounts.google.com/o/oauth2/token

Facebook example: https://graph.facebook.com/oauth/access_token

7. Configure the settings for each registered application. Application nodes and all child notes exist under the root node:
<Application>
</Application>

Update values for the following settings:

Application appname

Identifies the configuration for the OAuth registered application and the user authentication configuration. The end user must provide this identifier in the FCC page to use the configuration for the OAuth authentication.

Examples: googleapp, facebookapp

ApplicationURL

Specifies the registered application URL. Update the value of this setting with the same application redirect URL you entered during the application registration. Enter the URL using the following format:

https://cloudminder_host/siteminderagent/forms/oauthcb.fcc?SMQUERYDATA=Sample

Example:
https://cloud.ca.com/siteminderagent/forms/oauthcb.fcc?SMQUERYDATA=Sample

Important! If the SecureURLs parameter for the CAM-AgentObj object is set to "No", instead enter the ApplicationURL as follows:

http://cloudminder_host/siteminderagent/forms/oauthcb.fcc

By default, the SecureURLs parameter is set to "Yes" during SiteMinder Policy Server installation. You can check the value of the SecureURLs parameter in the CSP Console. Log in to the CSP Console, then click Infrastructure, Agent, Agent Configuration Objects. Select CAM-AgentObj and click Edit. Page forward to the SecureURLs parameter.

ClientID

Contains the identifier of the registered client application at the OAuth server. Update the value of this setting with the generated client ID. The authorization server provides this value when the application is successfully registered.

Secret

Indicates the secret associated with the ClientID. Update the value of this setting with the secret associated with the ClientID. The authorization server provides this value when the application is successfully registered.

PROVIDERLINK

Links the application with a provider. Specify the providername value of a defined provider. This application uses the provider configuration while performing OAuth authentication.

Examples: google, facebook

Scope

Specifies the required type of permission the application is requesting from the user. For example, if the scope value is https://www.googleapis.com/auth/userinfo.profile, the application can gain read-only access to basic user profile information.

This scope value is passed in the authorization token request. The client can use the code to access resource URLs, which are specified in the UserInfoURL attribute. Administrator can specify a single value or multiple space separated values for this attribute.

UserInfoURL

Designates a single URL or multiple space-separated URLs for which user information can be queried with the generated access token. The URL represents the resource the that client is trying access.

UserAttribute

Specifies a user attribute. Update this value with the user identifying claim from the OAuth user information. The value of this attribute is used to disambiguate the user. For Google or Facebook, set the user attribute to "email."

8. Save and close the file.

Customize the SPS Server Files for OAuth

A default OAuth Forms Credential Collector file (oauth-single.fcc) is included with the product. The FCC provides the list of OAuth application identifiers, information which is necessary for users to authenticate.

The path to the oauth-single.fcc file is entered in the authentication scheme configuration in the Administrative UI.

Note: In these procedures, *sps_home* refers to the Secure Proxy Server installation path. Secure Proxy Server is part of the CA CloudMinder product.

Follow these steps:

1. Log in to the cloud host system.
2. Navigate to the following location:

   sps_home/proxy-engine/examples/siteminderagent/forms

   sps_home

   Specifies the web Secure Proxy Server installation path. Secure Proxy Server is part of the CA CloudMinder product.

   Example: /opt/CA/secure-proxy/proxy-engine/examples/siteminderagent/forms.

3. Duplicate the oauth-single.fcc file and name the duplicate to reflect which OAuth provider, **and** which tenant, the file is for. For example:
   - oauth-google-tenant1.fcc
   - oauth-facebook-tenant2.fcc
4. Modify the renamed file and make sure that the hidden input value is set to the application name specified in the duplicated oauthproviders.xml file (see Copy and Modify the OAuth Provider Configuration File (see page 143)). For example:

   <input type="HIDDEN" name="oauth_appname" value="googleapp">

   <input type="HIDDEN" name="oauth_appname" value="facebookapp">
5. Save the file.
6. Navigate to the following folder:

   sps_home/Tomcat/webapps/affwebservices/redirectjsp

   Example: /opt/CA/secure-proxy/Tomcat/webapps/affwebservices/redirectjsp
7. Duplicate the oauth.jsp file and name the duplicate to reflect the OAuth provider **and** the tenant. For example:
   - oauth-google-tenant1.jsp
   - oauth-facebook-tenant2.jsp

Copy and Modify the OAuth Properties File

The OAuth properties file contains paths to the FCC file and the provider configuration file. The file is necessary for successful OAuth configuration. This file resides on the Policy Server.

Note: The OAuth authentication scheme configuration references this file.

Follow these steps:

1. Log in to the Policy Server host system.
2. Go to the following location:

   siteminder_home/config/properties

   siteminder_home

   Specifies the Policy Server installation path.

   Example: /opt/CA/siteminder/config/properties

3. Copy the oauth.properties file and name the copy to reflect the OAuth provider **and** the tenant.

   Examples:

   - oauth-google-tenant1.properties
   - oauth-facebook-tenant1.properties

4. In the file copy, change the following settings:

   FCC=/siteminderagent/forms/*duplicate_oauth_fcc_file*.fcc

   OAuthProviders=*oauthproviders_xml_file_path/duplicate_oauthproviders_file*.xml

   *Duplicate_oauth_fcc_file.*fcc is the Secure Proxy Server file you duplicated and renamed in the Customize the SPS Server Files for OAuth (see page 147) topic.

   *Duplicate_oauthproviders_file.*xml is the Policy Server file you duplicated and renamed in the Copy and Modify the OAuth Provider Configuration File (see page 143) topic.

   Google Example:

   FCC=/siteminderagent/forms/oauth-google-tenant1.fcc

   OAuthProviders=/opt/CA/siteminder/config/properties/oauthproviders-tenant1.xml

   Facebook Example:

   FCC=/siteminderagent/forms/oauth-facebook-tenant2.fcc

   OAuthProviders=/opt/CA/siteminder/config/properties/oauthproviders-tenant2.xml

5. (Optional). If the Policy Server host system is behind the proxy, set the ProxyAuthentication value to “yes" and specify the proxy details for the remaining settings. The password that you specify for the proxy user is used in the Secret field of the authentication scheme configuration.

   ProxyAuthentication=yes

   ProxyServer=

   ProxyPort=

   ProxyDomain=

   ProxyUser=

6. Set the PreProcessingChain setting to the following value:

   PreProcessingChain=com.ca.sm.oauth.chain.StoreClaimsToContext

7. Leave the AnonymousMode set to false.

   AnonymousMode=false

Copy and Modify the Open Format Expression File (Optional)

The openformatexpression.conf file enables OAuth self-registration. To configure this file, first generate an encrypted password key. The system uses the password and key during the self-registration process using OAuth. Then, modify the openformatexpression.conf properties file.

Note: You need to configure the openformatexpression.conf file only if you want to enable self-registration for your environment.

Follow these steps:

1. Log in to the Policy Server host system.
2. Go to your base Policy Server installation path.

   Example: /opt/CA/siteminder/

   Where siteminder is the folder where the Policy Server is installed.
3. Enter the following:

   source ca_ps_env.ksh
4. Go to the bin folder.
5. Enter the following:

   ./OpenFormatEncPwd.sh *password*

   where *password* is a password that you select

   The system returns an encrypted value for the password you enter. Write down the password you chose, and the exact encrypted value.
6. Navigate to the following location:

   siteminder_home/config/properties

   siteminder_home

   Specifies the Policy Server installation path.

   Example: /opt/CA/siteminder/config/properties
7. Copy the openformatexpression.conf file and name the copy to reflect the tenant.

   Examples:

   - openformatexpression-tenant1.conf
   - openformatexpression-tenant2.conf

8. In the file copy, add or modify the following settings:

   EncryptionTransform=AES256/CBC/PKCS5Padding

   EncryptionKey=<*encrypted password value*>

   SessionStore=false

   Prefix=SM_

   claim_given_name=first_name,given_name

   claim_family_name=last_name,family_name

   claim_email=mail,email

   claim_name=email,name

   TimeToLive=300

   Prefix=SMAUTHOAUTH_

   claim_ID=ID

   claim_name=MAIL,EMAIL,USERNAME,NAME

   encrypted password value

   Is the exact encrypted value that you created previously.

9. Save and close the file.

Set Openformat Cookie Properties (Optional)

Configure the open format cookie properties in your tenant environment to enable self-registration. You configure these properties in the CA CloudMinder Management Console.

Note: You need to configure the open format cookie properties only if you want to enable self-registration for your environment.

Follow these steps:

1. Log in to the Management Console.
2. Click Environments
3. Click the appropriate tenant name.

   The Environment Properties screen appears.
4. Click Advanced Settings, then Miscellaneous.

   The User Defined Properties screen appears.
5. Enter the following property and value pairs. After you enter each Property and Value, click Add.

Property	Value	Details
openformat.cookie.domain	*example.com*	Enter your environment domain name.
openformat.cookie.zone	SM	
openformat.cookie.name	DEFAULT	
openformat.cookie.encryption.password	*password*	Enter the clear-text password you used when you created the password encryption key for the openformatexpression.conf file. See Modify the Open Format Expression File (see page 150) for more information.
openformat.cookie.encryptiontype	AES256/CBC/PKCS5Padding	This is the encryption algorithm used by the OpenFormatEncPwd.sh tool. You used this tool to create the password encryption key for the openformatexpression.conf file.

6. Click Save.

   A warning appears that instructs you to restart the environment.

7. Click Restart Environment.

   The system saves the properties you added.

Configure the Custom OAuth Authentication Scheme

Follow these steps:

1. In the Administrative UI, click Infrastructure, Authentication.
2. Click Authentication Schemes.

   The Authentication Schemes page appears.

3. Click Create Authentication Scheme.

   Verify that the Create a new object of type Authentication Scheme is selected.

4. Click OK

   The Create Authentication Scheme page appears.

   Note: Click Help for descriptions of settings and controls, including their respective requirements and limits.

5. Enter a name and protection level. Do not use spaces in the name.

   Examples:

   - oauth_google_scheme
   - oauth_facebook_scheme

6. Select Custom Template from the Authentication Scheme Type pull-down menu.
7. Configure the scheme-specific settings:

   Library

   smjavaapi

   Secret/Confirm Secret

   If your system is behind a proxy server, enter the proxy server password. The associated user name must be provided in the oauth.properties file. The system uses these credentials to access external sites and gain access to the token and user information.

Parameter

Specify the OAuth authentication scheme implementation class name and the authentication scheme name and path to the OAuth properties file. The syntax is:

com.ca.sm.oauth.SmAuthOAuth <*auth_scheme_name*> <*oauth.properties_file_path*>

Examples:

com.ca.sm.oauth.SmAuthOAuth oauth_google_scheme opt/ca/siteminder/config/properties/oauth-google-tenant1.properties

com.ca.sm.oauth.SmAuthOAuth oauth_google_scheme ca/siteminder/config/properties/oauth-facebook-tenant2.properties

Note: Spaces separate the authentication scheme name and properties file path.

8. (Optional) Select the Persist Authentication Session Variables check box to persist authentication scheme claims in the session store.

   For persistent variables, the realm that uses this authentication scheme must support persistent sessions and the Policy Server must be configured with a session store.

9. Save the authentication scheme.

Enable Oauth Authentication Method for Tenant Environment

Before users can sign in using OAuth authentication servers, enable the OAuth authentication method for the entire tenant environment. You only need to enable Oauth once per environment. You can then configure any application in that environment to use OAuth authentication.

Follow these steps:

1. Log in to the User Console with application management privileges.

   For example, the default Tenant Administrator role has the appropriate privileges.
2. From the navigation menu, select Applications.
3. Click Authentication Methods, then Modify Authentication Method.

   A search screen appears.
4. Click Search.
5. Select the Google Authentication Method, and click Select.

   The Modify Authentication Method for Google screen appears.
6. Select the Enabled check box.
7. In the Authentication Method Scheme drop-down list, select Google.
8. Update the Authentication URL field to the following:

   /affwebservices/*tenant_tag*/*duplicate_oauth_file*.jsp

   Tenant_tag is the unique identifier for a given tenant. *Duplicate_oauth_file*.jsp is the Secure Policy Server file you duplicated and renamed in the Customize the SPS Server Files for OAuth topic (see page 147).
9. If you want to enable self-registration, select the Enabled for Self Registration check box.

   Note: Additional steps are required to complete self-registration configuration. These steps are described in later topics in this scenario.
10. Click Submit.

Apply OAuth Authentication Method to Tenant

Apply the OAuth authentication method to your tenant. Users in a tenant environment can then sign-on to CA CloudMinder through OAuth authentication servers, such as Google and Facebook.

Follow these steps:

1. Log in to the User Console with application management privileges.

   For example, the default Tenant Administrator role has the appropriate privileges.

2. From the navigation menu, select Applications.
3. Click Applications, then Modify Application.

   The Modify Application search screen appears.

4. Search for the tenant for which you want to enable Oauth.

   To display a list of all tenants for which you have administrative privileges, click Search without modifying the search criteria.

5. Select the appropriate tenant and click Select.

   The Modify Application screen appears.

6. In the Authentication Methods area, click Add.

   The Select Authentication Methods search screen appears.

7. Click Search.

   A list of authentication methods available in your environment appears.

8. Select Google, and click Select.
9. Click Submit.

Use the Authentication Scheme in a Policy

For SSO applications, you must establish a one-to-one correspondence between an authentication method configured at the User Console and an authentication scheme configured at the Administrative UI. The authentication method and the scheme work together to enforce user authentication for a requested SSO application.

After you create an authentication scheme, the scheme has to protect the authentication URL specified for a given authentication method. To protect the URL, the scheme is assigned to a realm, and the realm becomes part of a policy.

Follow these steps::

1. Configure the policy domain for the tenant.
2. Assign user directories to the tenant domain.
3. Create a realm and rule for the tenant domain.
4. Create a policy to protect the authentication URL.

Select the Policy Domain for the Tenant

A policy domain protects logical groupings of resources. When you deploy a tenant in the Administrative UI, the system automatically generates a policy domain for the tenant.

Work with this domain when establishing a policy for the authentication URL.

Follow these steps:

1. In the Administrative UI, click Policies, then click Domain.
2. Click Domains.

   The Domains page appears.
3. Select the domain for the appropriate tenant and modify it.
4. Confirm that the tenant user directory is part of the tenant domain. If it is not, add the user directory to the domain (see page 121).
5. Click Submit.

   The policy domain is modified.

Assign User Directories to the Tenant Domain

You can add one or more tenant user directories to a tenant domain. The system authenticates users by comparing the user credentials to the credentials that are stored in the user directories. The system searches the user directories in the same order that they are listed in the policy domain.

Note: The following procedure assumes that the tenant user directory is already configured in the Administrative UI.

Follow these steps:

1. Click Policies, Domain.
2. Click Domains.

   The Domains page appears.
3. Specify the search criteria and click Search.

   A list of domains that match the search criteria appears.
4. Click the name of the domain that you want to modify.

   The View Domain page appears.
5. Click Modify.

   The settings and controls become active.

6. In the General tab, click Add/Remove.

   The Choose user directories page appears.

7. Select one or more user directories from the list of Available Members, and click the right-facing arrows.

   The user directories are removed from the list of Available Members and added to the list of Selected Members.

   Note: To select more than one member at one time, hold down the Ctrl key while you click the additional members. To select a block of members, click the first member then hold down the Shift key while you click the last member in the block.

8. Click OK.

   The selected user directories are listed under User Directories.

   Note: To create a user directory and add it to the domain, click Create.

9. Click Submit.

   The selected user directories are added to the domain.

Configure a Realm and a Rule for the Tenant Domain

A realm groups resources that have similar security requirements and share a common authentication scheme. For the tenant domain, create a realm and associate it with a Web Agent.

Note: The following procedure assumes that you are creating an object. You can also copy the properties of an existing object to create an object.

Follow these steps:

1. Click Policies, Domain, Realms.

   The Realms page appears.

2. Click Create Realm.

3. Select the tenant domain that you want to modify, and click Next.

4. Type the name and a description of the realm.

   Specify a name that indicates the realm is for an SSO authentication URL. For example:

   - OAuth_Google_Realm
   - OAuth_Facebook_Realm

5. Click Lookup Agent/Agent Group to select an agent.

6. Select the **cam-agent** and click OK.

7. Specify the Resource Filter for the authentication scheme you are using. This scheme has to tie in to the authentication method chosen in the User Console configured and applied to the application.

   The following list includes the resource filter for all available authentication schemes for cloud SSO. Use the resource filter for your authentication scheme.

   For HTML Forms authentication

   For environments created in CA CloudMinder 1.51 or later:

   /chs/redirect/*tenant_tag*/forms

   For environments created before CA CloudMinder 1.51:

   /affwebservices/<tenant-name>/forms.jsp

   For OpenID authentication

   /affwebservices/*tenant_tag*/*duplicate_openid_file*.jsp

   Copy the default openid.jsp file to a unique name, such as openid-google.jsp. Having a unique jsp file is necessary to distinguish openID configurations.

   For OAuth authentication

   /affwebservices/*tenant_tag*/*duplicate_oauth_file*.jsp

   Copy the default oauth.jsp file and give the copy a unique name, such as oauth-google.jsp or oauth-facebook.jsp. Having a unique jsp file is necessary to distinguish OAuth configurations.

   For Arcot PKI authentication scheme

   For environments created in CA CloudMinder 1.51 or later:

   /chs/redirect/*tenant_tag*/arcotid

   For environments created before CA CloudMinder 1.51:

   /affwebservices/<tenant-name>/arcotid.jsp

   For Arcot OTP authentication scheme

   For environments created in CA CloudMinder 1.51 or later:

   /chs/redirect/*tenant_tag*/arcototp

   For environments created before CA CloudMinder 1.51:

   /affwebservices/<tenant-name>/arcototp.jsp

For Arcot PKI Risk authentication scheme

For environments created in CA CloudMinder 1.51 or later:

/chs/redirect/*tenant_tag*/arcotidrisk

For environments created before CA CloudMinder 1.51:

/affwebservices/<tenant-name>/arcotidrisk.jsp

For Arcot OTP Risk authentication scheme

/chs/redirect/*tenant_tag*/arcototp_risk

For environments created in CA CloudMinder 1.51 or later:

/chs/redirect/*tenant_tag*/arcototprisk

For environments created before CA CloudMinder 1.51:

/affwebservices/<tenant-name>/arcototprisk.jsp

tenant_tag is a unique identifier for a tenant. You specify the tag when deploying a tenant environment in the Administrative UI. To view a list of tags, select the Tenants tab.

8. Complete the remaining fields:

   Default Resource Protection

   Protected

   Authentication Scheme

   Select the authentication scheme that you configured for the SSO application and the scheme that corresponds to the resource filter. For example, if you are using OpenID and you configured a scheme named OpenID Auth, select that scheme.

9. Create a rule:

   a. Specify a name for the rule.

      For example, if Google is the OAuth provider, name the rule oauth_googlerule. If Facebook is the OAuth provider, name the rule oauth_facebookrule.

   b. In the Realm and Resource area, edit the Resource value by deleting the forward slash (/) character.

      Important! The Resource value is now the asterisk (*) character only.

   c. In the Action area, select Web Agent actions.

   d. Under Action, control-click to multi-select GET, HEAD, and POST.

   e. Accept the defaults for the remaining settings.

   f. Click Ok.

10. Specify the session properties.

    Note: Click Help for a description of fields, controls, and their respective requirements.

11. Skip the other configuration options.
12. Click Finish.

    The realm is complete.

Create the Policy to Protect the Authentication URL

Create a policy for the domain. Policies define relationships between users and resources. The policy components work together and protect the resource.

After you create the policy, you add users and rules.

Follow these steps:

1. Click Policies, Domain.
2. Click Domains.
3. Specify search criteria, and click Search.

   A list of domains that match the search criteria appears.

4. Click the edit icon next to the domain for which you want to create a policy.

   The Modify Domain page appears.

5. Click the Policies tab.

   The Policies page appears.

6. Click Create.

   The Create Policy page appears.

7. Enter a name and a description for the policy. Use a name that indicates that the policy is for the authentication URL.
8. Add individual users, user groups, or both from the Users tab. The users are members of the tenant user directory associated with the domain. When a user tries to access a protected resource, the policy verifies whether the user is allowed to access the resource.

   Note: If you select Add Members, the User/Groups pane opens. Individual users are not displayed automatically. Use the search utility to find a specific user within one of the directories.

   You can edit or delete a user or group by clicking the right arrow (>) or minus sign (-), respectively.

9. When you have finished selecting users, user groups or both, click OK.

10. Add rules to the policy from the Rules tab.

    Rules indicate which resources are part of a policy and whether to allow or deny access to the resources.

    Note: Add at least one rule or rule group to a policy.

    The Available Rules pane opens.

11. Select the rule that you created for the authentication URL resource.

    For example, if you configured a rule specific to Google, named oauth_googlerule, select that rule.

    You are not required to configure a response for the rule.

12. Click OK.
13. Click Submit to save the policy.

The policy configuration is complete.

Complete OAuth Self-Registration Configuration

For SSO applications, you must establish a one-to-one correspondence between an authentication method configured at the User Console and an authentication scheme configured at the Administrative UI. The authentication method and the scheme work together to enforce user authentication for a requested SSO application.

After you create an authentication scheme, the scheme has to protect the authentication URL specified for a given authentication method. To protect the URL, the scheme is assigned to a realm, and the realm becomes part of a policy.

Follow these steps::

1. Configure the policy domain that represents the tenant.
2. Assign user directories
3. Create a realm.
4. Create a policy to protect the authentication URL.

Create a Rule for Self-Registration

Create a self-registration rule for your tenant.

Follow these steps:

1. In the Administrative UI, click Policies, then click Domain.
2. Click Realms.

   The Realms page appears.
3. Click the Edit icon to modify the OAuth realm that you created for this OAuth provider and tenant.

   You created this realm in the step entitled Configure a Realm and Rule for the Tenant Domain (see page 122).

   The Modify Realm screen appears.
4. In the Rules area, click Create.

   The Create Rule screen appears.
5. Enter a name and description for the rule. Name the rule to reflect the authorization scheme, provider **and** tenant, for example:
   - OAuth_Google_SelfReg_Tenant1
   - OAuth_Facebook_SelfReg_Tenant1
6. In the Realm and Resource area, edit the Resource value by deleting the forward slash (/) character.

   Important! The Resource value is now the asterisk (*) character only.
7. In the Action area, select Authentication events.
8. In the Action drop-down, select OnAuthAttempt.
9. Accept the defaults for the remaining settings.
10. Click Ok.
11. Click Submit.

    The system creates the self-registration rule for your tenant.

Create a Response for Self-Registration

Create a self-registration response for your tenant.

Follow these steps:

1. In the Administrative UI, click Policies, then click Domain.
2. Click Responses.
3. Click Create Response.

   The Create Response: Select Domain screen appears.
4. Select the tenant domain for which you are configuring OAuth, and click Next.

   The Create Response: Define Response screen appears.
5. Enter a name and description for the response. Name the response to reflect the authorization scheme, provider **and** tenant, for example:
   - OAuth_Google_SelfReg_Response_Tenant1
   - OAuth_Facebook_SelfReg_Response_Tenant1
6. Click Create Response Attribute

   The Create Response Attribute screen appears.
7. In the Attribute Type Area, under the Attribute drop-down, select:

   WebAgent-OnReject-Redirect
8. In the Attribute Setup area, enter the following URL in the Variable Value field:

   https://<*hostname*>/iam/im/<*tenantpublicalias*>/ui7/index.jsp?task.tag=CAMSelfRegistrationOpenFormat

   <*hostname*> is your tenant domain. <*tenantpublicalias*> is the public name of your tenant domain.

   If you do not know the public alias for your tenant, log in to the Management Console, click Environments, then click the tenant environment you want. Copy the tenant Public Alias and enter it in the above URL in place of <*tenantpublicalias*>.
9. Click Ok.

   The Create Response: Define Response screen appears, updated with the attribute you created.
10. Click Create Response Attribute

    The Create Response Attribute screen appears.
11. In the Attribute Type Area, under the Attribute drop-down, select:

    WebAgent-HTTP-Cookie-Variable
12. In the Attribute Setup area, under Attribute Kind, select Active Response.
13. In the Attribute Setup area, under Attribute Fields, enter the following values:

- For Cookie Name, enter SMDEFAULT
- For Library Name, enter smjavaapi
- For Function Name, enter JavaActiveExpression
- For Parameters, enter:

  com.ca.sm.expression.activeresponse.OpenFormatCookieExpression /opt/CA/siteminder/config/properties/*<openformatexpression_file.conf>*

  <openformatexpression_file.conf> is the Open Format Expression file you duplicated and renamed in the Copy and Modify the Open Format Expression File (see page 150) topic.

  Note: A space separates the library name, "com.ca.sm.expression.activeresponse.OpenFormatCookieExpression" and the open format expression configuration file path, "/opt/CA/siteminder/config/properties/*<openformatexpression_file.conf>"*

Note: As you enter the previous values, the values are reflected in the Script field. When you have finished entering these values, the Script field reads:

SMDEFAULT=<@lib="smjavaapi" func="JavaActiveExpression" param="com.ca.sm.expression.activeresponse.OpenFormatCookieExpression /opt/CA/siteminder/config/properties/*<openformatexpression_file.conf>*" @>

14. Click Ok.

    The Create Response: Define Response screen appears, updated with the attribute you created.

15. Click Finish.

    The system creates the self-registration response for your tenant.

Add Self-Registration Rule and Response to the Policy

Add the rule and response you created for self-registration to the tenant policy.

Follow these steps:

1. In the Administrative UI, click Policies, then click Domain.
2. Click Domains.

   A list of tenant domains appears.
3. Click the edit icon next to the tenant domain for which you are configuring self-registration.

   The Modify Domain screen appears.
4. Click the Policies tab.
5. Click the edit icon next to the OAuth policy you created for this tenant.

   You created this policy in the step entitled Create the Policy to Protect the Authorization URL (see page 125).
6. Click the Rules tab.
7. Click Add Rule

   A list of available rules appears.
8. Select the self-registration rule you created for this tenant.

   You created this rule in the step entitled Create a Rule for Self-Registration (see page 163).
9. Click Ok.
10. Click Add Response next to the self-registration rule you just added.

    A list of available responses appears.
11. Select the self-registration response you created for this tenant.

    You created this response in the step entitled Create a Response for Self-Registration (see page 164).
12. Click Ok.

    The Modify Policy screen appears, updated with the self-registration rule and response you added.
13. Click Ok.
14. Click Submit.

    The configuration process for enabling OAuth with self-registration is now complete.

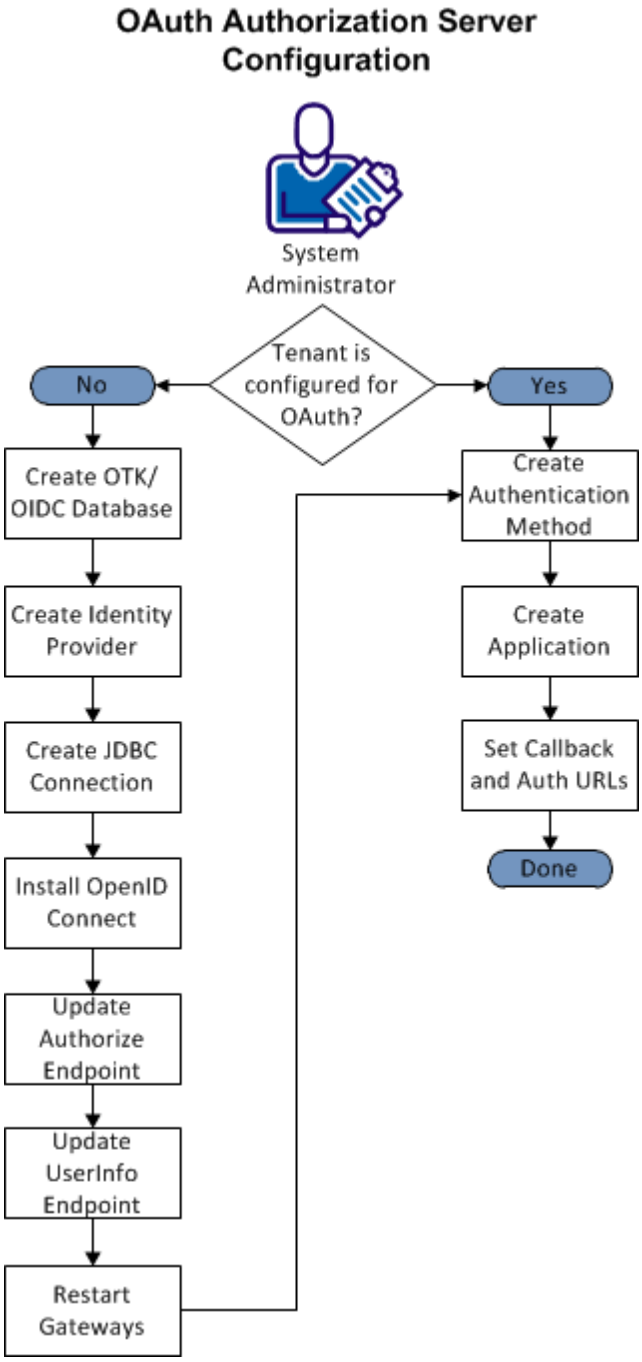
Chapter 5: SSO Using CloudMinder as an OAuth Authorization Server

As a CSP administrator, you can configure single sign-on so that CA CloudMinder acts as an OAuth authorization server for an OAuth client. You perform configuration steps for each tenant where SSO for an OAuth client is required. You perform these steps only once per tenant.

You also perform configuration steps for each individual OAuth client that will use CA CloudMinder as an OAuth authorization server. You perform these steps for each OAuth client that is associated with the tenant.

For example, the tenant Forward, Inc. wants to use CA CloudMinder as an OAuth authorization server for to two web application and a mobile application. Each of these applications use OAuth for authentication communication; they are called *OAuth clients*. As a CSP administrator, you configure CloudMinder as an OAuth authorization server for the Forward, Inc. tenant environment. Then you configure single sign-on using OAuth for the two web applications and the mobile application. Because each application has specific callback and authentication URLs, you configure each one separately.

The following figure illustrates the required configuration tasks.



The following procedures describe each task in detail.

Do the following for each tenant where SSO for an OAuth client is required. Perform these steps only once per tenant. Begin here if you have not configured the OAuth authorization server for this tenant yet.

1. Create the OTK/OIDC Database (Oracle) (see page 170) or Create the OTK/OIDC Database (PostgreSQL) (see page 171)
2. Create an Identity Provider for CA Directory (see page 172)
3. Create a JDBC Connection to the OTK/OIDC Database (Oracle) (see page 174) or Create a JDBC Connection to the OTK/OIDC Database (PostGres) (see page 175)
4. Install OpenID Connect (see page 176)
5. Update the Authorize Endpoint (see page 177)
6. Update the UserInfo Endpoint (see page 178)
7. Restart Gateways (see page 180)

Do the following for each OAuth client you want to use the OAuth authorization server. Perform these steps for every OAuth client. Begin here if you have already configured the OAuth authorization server for this tenant, and you want to configure individual OAuth clients to use it.

1. Create the Authentication Method (see page 126)
2. Create an Application (see page 58)
3. Set Callback and Authentication URLs (see page 186)

Create the OTK/OIDC Database (Oracle)

Create the Layer 7 Oracle tablespaces, and an Oracle user, for your tenant.

Note: Perform this procedure for each tenant for which you configure CA CloudMinder as an external IdP using OAuth. Perform this procedure only once for each tenant, even if the tenant has many OAuth clients.

Follow these steps:

1. Connect to the CloudMinder Oracle database as the system user.

   For example, connect with SQL Developer.

2. Open the following file:

   oracle_oidc.sql

3. Save a copy of the file with a name that is specific to the tenant for which you are configuring OAuth.

   For example:
   oracle_oidc_forwardinc.sql

4. Replace all instances of <OTKDB-USERNAME> with a user name you choose. Choose a user name that is unique to the current tenant.

5. Replace <OTKDB-PASSWORD> with a password you choose.

   Note: Make a note of the username and password for use later in the configuration process. For an Oracle RAC installation, specify the data file based on the Disk Group Name.

6. For an Oracle RAC installation, modify the Oracle script to specify a disk group name as defined in the Oracle environment. Replace the existing CREATE TABLESPACE query with a query in a form similar to this example:

   ```
   CREATE TABLESPACE tbs_<OTKDB-USERNAME>
    DATAFILE '+DATA'
      SIZE 10M
      REUSE
      AUTOEXTEND ON NEXT 10M MAXSIZE 200M;
   ```

7. Save and close the file.

8. Execute the modified version of the script.

9. Click Commit.

Create the OTK/OIDC Database (PostgreSQL)

This section applies for tenant deployments where the OTK/OIDC database is PostgreSQL-driven.

Note: Perform this procedure for each tenant for which you configure CA CloudMinder as an external IdP using OAuth. Perform this procedure only once for each tenant, even if the tenant has many OAuth clients.

Follow these steps:

1. Connect to the CA CloudMinder PostgreSQL database as the system user.

   For example, connect with the psql client.

2. Open the following file:

   postgres_oidc.sql

3. Save a copy of the file, using a filename that represents the tenant for which you are configuring OAuth. For example:

   postgres_oidc_forwardinc.sql

4. Replace all instances of <OTKDB-USERNAME> with a user name that is unique to the current tenant.

5. Replace all instances of <OTKDB-DB> with a database name that is unique to the current tenant.

6. Replace <OTKDB-PASSWORD> with a password you choose.

7. Make a note of the username and password for use later in the configuration process.

8. Save and close the file.

9. Execute the modified version of the script.

10. Click Commit.

Create an Identity Provider for CA Directory

Create an Identity Provider for CA Directory, which acts as the OpenID UserInfo Endpoint. This allows CloudMinder to act as an external Identity Provider for applications you want to authenticate via OAuth.

Note: Perform this procedure for each tenant for which you configure CA CloudMinder as an external IdP using OAuth. You only need to do this once per tenant, not once per OAuth client for that tenant.

Follow these steps:

1. Navigate to the Layer 7 Policy Manager web interface at the following URL:

   https://<GATEWAY_ONE_HOSTNAME>:8443/ssg/webadmin

2. Log in using the credentials that were created during installation for the Gateway admin user.
3. In the upper-left pane, click the Identity Providers tab.
4. In main window area, click Create LDAP Identity Provider.

   The Create LDAP Identity Provider Wizard opens.

5. Under Provider Type, select GenericLDAP.
6. In the Provider Name field, enter a meaningful name for the CA Directory Identity Provider.

   For example, enter CA Directory.

7. In the LDAP Host URL field, enter the following:

   ldap://<*LOAD BALANCER VIP*>:20498

   The DxRouter instances to which the Layer 7 Gateway connects run on the same machine as the SiteMinder Policy Server. For <*LOAD BALANCER VIP*>, enter the VIP of the SiteMinder Policy Server used on the application tier load balancer.

   20498 is the is the LDAP port on which the Gateway is listening. Use this port number unless you have changed the LDAP port.

8. In the Search Base field, enter your LDAP search root.

   For example, ou=xxx,ou=xxx,o=xxx

   To locate this information, log in to the Administrative UI, click Infrastructure, then Directory, then User Directories. Click to view the User Directory for your tenant. Under LDAP Search, the values labeled Root are your LDAP search root. Copy and Paste these values into the Search Base field.

9. In the Bind DN field, enter your bind DN.

   For example, cn=xxx,ou=xxx,ou=xxx,o=xxx

   To locate this information, log in to the Administrative UI, click Infrastructure, then Directory, then User Directories. Click to view the User Directory for your tenant. Under Administrator Credentials, the values labeled Username are your Bind DN. Copy and Paste these values into the Bind DN field.

10. In the Bind Password field, enter your bind password.

    This is the LDAP database connection password, specified during creation of the LDAP server. If you do not know this password, see your LDAP administrator.

11. Click Test to verify the connection.

12. Click Ok.

13. Click Finish.

Create a JDBC Connection to the OTK/OIDC Database (Oracle)

If the OTK/OIDC database is Oracle-driven, use the following procedure to create a JDBC connection to the database. This connection enables the Layer 7 Gateway to exchange authentication and authorization data with CA CloudMinder.

Note: Perform this procedure for each tenant for which you configure CA CloudMinder as an external IdP using OAuth. Perform this procedure only once for each tenant, even if the tenant has many OAuth clients.

Follow these steps:

1. In the Layer 7 Policy Manager web interface, click Manage, then Manage JDBC Connections.
2. Click Add.
3. Enter a meaningful Connection Name that corresponds to your JDBC connection.

   For example, enter:

   Forward Inc. JDBC
4. In the Driver Class field, enter the following:

   com.l7tech.jdbc.oracle.OracleDriver
5. Complete the JDBC URL field:

   For a standalone Oracle installation, enter the following:

   jdbc:l7tech:oracle://<*DB-HOSTNAME*>:<*DB-PORT*>;Database=<*DB*>

   Where <*DB-HOSTNAME*> is the hostname for the Oracle database, <*DB-PORT*> is the port on which the database is listening, and <*DB*> is the name of the Oracle service.

   For an Oracle RAC installation, enter the following:

   jdbc:l7tech:oracle://<DB-HOSTNAME>:<DB-PORT>;ServiceName=<DB>
6. In the User Name field, enter the user name that you specified for <OTKDB-USERNAME> in the oracle_oidc.sql file for this tenant.

   You created this user name, and the password that is referenced in the following step, in the procedure entitled the Create the OTK/OIDC Database (Oracle) (see page 170).
7. In the Password field, enter the password that you specified for <OTKDB-PASSWORD> in the oracle_oidc.sql file for this tenant.
8. Click Test to verify the connection.
9. Click OK, then OK.

Create a JDBC Connection to the OTK/OIDC Database (PostGres)

If the OTK/OIDC database is PostgreSQL-driven, use the following procedure to create a JDBC connection to the database. This connection enables the Layer 7 Gateway to exchange authentication and authorization data with CA CloudMinder.

Note: Perform this procedure for each tenant for which you configure CA CloudMinder as an external IdP using OAuth. Perform this procedure only once for each tenant, even if the tenant has many OAuth clients.

Follow these steps:

1. In the Layer 7 Policy Manager web interface, click Manage, then Manage JDBC Connections.
2. Click Add.
3. Enter a meaningful Connection Name that corresponds to your JDBC connection. For example, enter the following:

   Forward Inc. JDBC
4. In the Driver Class field, enter the following:

   org.postgresql.Driver
5. In the JDBC URL field, enter the following:

jdbc:postgresql://<DB-HOSTNAME>:<DB-PORT>/<OTK-DB>

Where <DB-HOSTNAME> is the hostname for the PostgreSQL database, <DB-PORT> is the port on which the database is listening, and <OTK-DB> is the name of the database that was specified in the postgres_oidc.sql file for this tenant.

1. In the User Name field, enter the user name you specified for <OTKDB-USERNAME> in the postgres_oidc.sql file for this tenant.
2. In the Password field, enter the password you specified for <OTKDB-PASSWORD> in the postgres_oidc.sql file for this tenant.
3. Click Test to verify the connection.
4. Click OK, then OK.

Install OpenID Connect

Installing OpenID Connect allows the Layer 7 Gateway to act as an Open ID User Information Endpoint. During an authentication request, a client can request additional user profile information. As a User Information Endpoint, the Layer 7 Gateway can fulfill that request.

Note: Perform this procedure for each tenant for which you configure CA CloudMinder as an external IdP using OAuth. You only need to do this once per tenant, not once per OAuth client for that tenant.

Follow these steps:

1. In the lower-left pane of the Policy Server interface, select the folder where you want to install policies.

   We recommend that you install in the root folder. Select the root node that is labeled with the hostname of the Gateway.

2. Select Manage, then Additional Actions, then Install OpenID Connect.

3. For Prefix, enter the protected alias you specified when you created this tenant. The prefix identifies the particular tenant in the CloudMinder environment for which you are configuring OAuth authentication.

   To locate the protected alias for the tenant, log in to the Management Console, click Environments, then click to view the environment for the current tenant. The protected alias for the tenant is listed.

   Note: In a single-tenant Cloudminder deployment, such as a test or development environment, you can leave the Prefix field blank.

4. Select Core Services and Test Client.

5. Under Map Policy JDBC Connection, select the JDBC connection you created previously.

6. Click Install.

   When install is complete, the system creates a subfolder under root entitled "MAG-2.0" and the tenant prefix you entered.

Update the Authorize Endpoint

Note: Perform this procedure for each tenant for which you configure CA CloudMinder as an external IdP using OAuth. You only need to do this once per tenant, not once per OAuth client for that tenant.

Follow these steps:

1. In the lower-left pane of the Policy Server interface, enter the following into the Search field:

   authorize

2. If more than one search result is returned, select the one that ends with <PREFIX>/auth/oauth/v2/authorize in brackets. For example:

   OAuth 2.0/oauth/v2 [<PREFIX>/auth/oauth/v2/authorize]

3. Double-click to open the policy assertions for this endpoint.

   The list of assertions for this endpoint appear.

4. In the policy assertion pane, enter the following into the Search field:

   CHANGEME

   The system highlights the appropriate assertion.

5. Double-click to open the assertion.

6. Set the siteminder.resource context variable to the known protected resource path. The path has the following format:

   /chs/redirect/*tenant*/forms

   To locate the protected resource path for the tenant:

   a. Log in to the Administrative UI.

   b. Click Policies, then Domain.

   c. In the left-hand menu, click Realms.

   d. Click to open the <TENANT>_chsforms_realm_es realm.

      Where <TENANT> is the name you assigned your tenant upon creation. Keep in mind that ten realms are listed per page, and that this realm may be on a subsequent page.

   e. The Resource Filter is your protected resource path. Copy and paste this value into the Expression field for the siteminder.resource context variable in the Layer 7 Policy Server.

7. Click OK.

8. Click Save and Activate.

Update the UserInfo Endpoint

Note: Perform this procedure for each tenant for which you configure CA CloudMinder as an external IdP using OAuth. You only need to do this once per tenant, not once per OAuth client for that tenant.

Follow these steps:

1. In the lower-left pane of the Policy Server interface, enter the following into the Search field:

   userinfo

2. If more than one search result is returned, select the one that ends with <PREFIX>/openid/connect/v1/userinfo in brackets. For example:

   Protected endpoints/MSSO related/UserInfo
   [<PREFIX>/openid/connect/v1/userinfo]

3. Double-click to open the policy assertions for this endpoint.

   The list of assertions for this endpoint appear.

4. In the policy assertion pane, enter the following into the Search field:

   CHANGEME

   The system highlights the appropriate assertion.

5. Right-click the assertion, and select Enable Assertion from the menu.

6. Double-click to open the assertion.

7. Set LDAP Connector to the identity provider you created previously.

   You created the identity provider in the Create an Identity Provider for CA Directory (see page 172) topic.

8. Click OK.

9. Click Save and Activate.

Update the Tenant Web Services Fragment

Note: Perform this procedure for each tenant for which you configure CA CloudMinder as an external IdP using OAuth. Perform this procedure once per tenant, not once per OAuth client for the tenant.

The connection to the tenant web service (TWS) will need to be set up to fetch tenant information for the presentation of the OAuth grant page. The necessary values can typically be found inside the chsConfig.properties file on the Secure Proxy Server.

Follow these steps:

1. Add logo to the tenant configuration using the User Console. Expand Tenant Administration and click Tenant Settings.
2. In the lower-left pane of the Policy Server interface, enter the following into the Search field:

   tws
3. If more than one search result is returned, select the one that ends with <PREFIX>/…/tws/TWS Fetch Tenant Information. For example:

   Policy Fragments/tws/TWS Fetch Tenant Information
4. Double-click to open the policy assertions for this fragment.

   The list of assertions for this fragment appear.
5. In the policy assertion pane, enter the following into the Search field:

   CHANGEME

   The system displays 4 assertions that must be modified.
6. Change WS Fetch Tenant Information (the ChangeMe variables). The necessary values usually exist in the chsConfig.properties file on the Secure Proxy Server in this location:

   /opt/CA/secure-proxy/Tomcat/webapps/chs/WEB-INF/classes/config

   Note: The password in the config file is encrypted, but it has the decrypted value in the fragment.
   #Provide hostname and port of the deployed tenant-services application

   For example:
   tenantwebservicebaseurl=http://west_ex:9090/tenant-services/cm/tenantws

   #Provide shared secret key to authenticate client by tenant-services application

   Supply a clear text password, when prompted by Layer 7.

   #Provide tenant-services configuration Id

   For example: configurationid=tenantwebservice

   #connection time out in milliseconds. The value zero means timeout of infinity.
   #default value is 30 seconds if no value specified

For example:connection_timeout=30000

7. For each of the 4 assertions, double-click to supply these values for the context variables:
   - For tws.shared_secret, supply the unencrypted password for TWS.
   - For context variable tws.base_url, supply the base URL for TWS.
   - For context variable tws.configuration_id, supply the configuration ID for TWS
   - For context variable tws.tenant_name, supply the name of the tenant.
8. For each of the 4 assertions, click OK to close prompt.
9. Click Save and Activate.

Restart Gateways

To update configuration changes, restart the service on both Layer 7 Gateways. Run the following commands:

```
Service ssg stop
service ssg start
```

You have now configured your tenant to act as an OAuth authorization server. You need not perform these configuration steps again. You can now perform the configuration steps required for each OAuth client you want to connect to the OAuth authorization server.

Create an Application

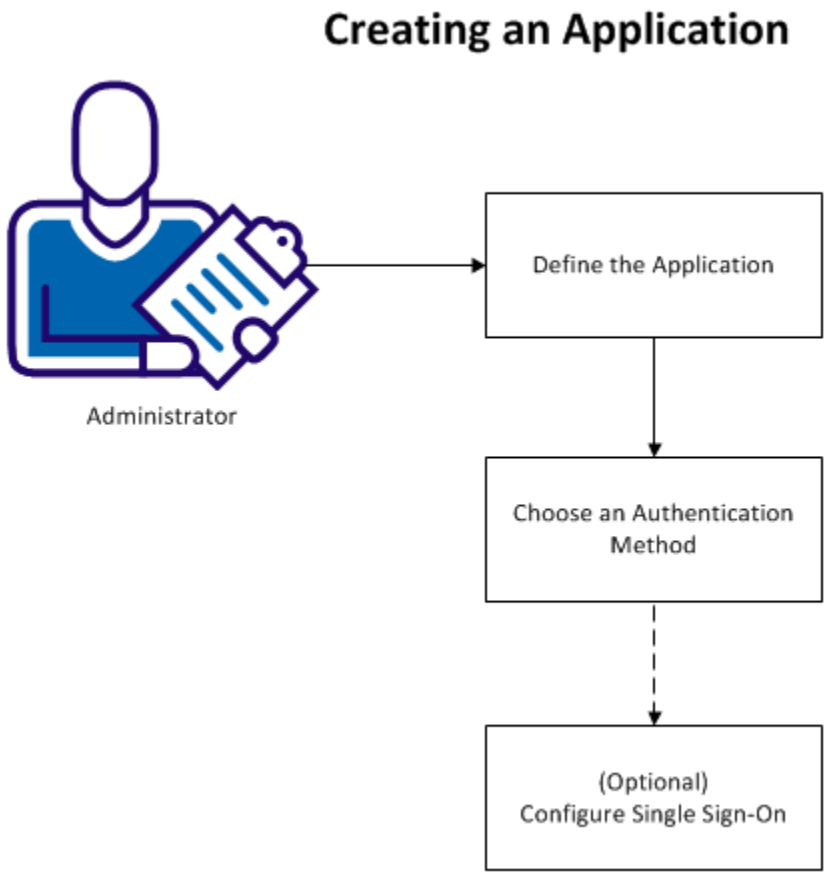
As an administrator, you want to give your users secure and convenient access to software resources. For example, your users need access to your email system, which can be hosted on-premise by your organization. Users also need access to Salesforce.com, which an external organization hosts in the cloud.

Note: Although a tenant administrator typically creates applications for their environment, a hosting administrator can also perform this task.

You create an application to define how users access a software resource. For example, when you configure an application, you define what type and level of security protects the resource. If you have purchased CA CloudMinder Advanced Authentication, you can configure advanced security such as two-factor authentication to protect the resource. If you have purchased the CA CloudMinder Single Sign-on service, you can configure SSO for the application. Users only log in once to access all applications that are configured for SSO.

Once an application is configured, you can give users access to the software resource. You can configure a service that includes the application, and can assign the service to users. The users can click the icon in the User Console Home Page to access the application. For more information, see Creating a Service Using the Service Wizard (see page 64). You can also give users access to the application by making a link to the application available. For example, you can insert the link into any web page or you can send the link in an email.

The following diagram shows the information to understand and the steps to perform to create an application and make it available to users.



The following topics explain how to create an application:

1. Define the Application Profile (see page 59)
2. Choose an Authentication Method (see page 61)
3. (Optional) Configure Single Sign-On (see page 63)

Define the Application

You define the application details through the User Console.

Follow these steps:

1. Log in with an account that has application management privileges.

   For example, the default Tenant Administrator role has the appropriate privileges.

2. From the navigation menu, select Applications.

3. Click Applications, then Create Application.

   The Create Application screen appears.

4. Enter a name and description.

5. Associate a group with the application, if desired.

   Only the users who are members of the indicated group receive access.

   Note: If you are configuring the application for SSO access, the group that you choose must match the group name that is indicated in the SSO partnership configuration for this application. Only if the group names match will the system restrict access to group members. To confirm the group name that is indicated in the partnership configuration, refer to your hosting administrator. SSO partnership configuration information is available in the CSP Console.

6. Enter a launch URL for the Application.

   A launch URL is the fully qualified domain name of the software resource you want to make available to users. For example, if a user clicks the icon for this application in the User Console Home page, they are directed to the launch URL.

   If you are configuring the application for SSO access, the launch URL is the SSO Service URL generated during SSO partnership configuration. Refer to your hosting administrator for this information.

   If you are not configuring an SSO application, simply enter the fully qualified domain name of the software resource. Use the following format:

   https://softwareresourcedomainname.com

7. Choose a logo.

   This logo is the icon for the application that appears in the User Console Home page. Users can click the icon to access the software resource.

   Note: You can also give users access to the application by inserting a link to the application into any web page.

8. Enter a welcome message.

   When users click any link you provide to the application, a login screen appears. The welcome message appears at the top of this screen.

9. Select a self-registration task.

   If a user attempts to access the application but the user does not have a CA CloudMinder account, you can allow them to self-register. Choose one of the following self-registration tasks:

   Create New Account

   Presents a simple registration form. Upon submission, creates a user account.

   Create New Account with Workflow

   Presents a simple registration form. Upon submission, forwards the user account request to one or more approvers. Creates an account upon approval.

   Create New Account with Domain Validation

   Presents a simple registration form. Upon submission, compares the email domain of the user to the tenant email domain. If they match, sends a confirmation email to the user. Creates an account upon user confirmation.

   Note: The tenant email domain is specified in the User Console, under Tenant Administration, Tenant Settings.

   Self-Registration with Attribute Exchange

   Do not choose this self-registration task in the context of application access. This task is intended for a separate purpose.

10. Choose an authentication method. (see page 61)

Choose an Authentication Method

In the Create Application screen, continue the process of creating an application by choosing one or more authentication methods. When a user attempts to access the application, the system presents a login screen. The authentication methods that you choose appear on this screen. The user can log in using their choice of the available authentication methods.

For example, you can select the Basic and Google External IDP authentication methods for an application. The application login screen displays user name and password fields for basic authentication. The login screen also displays the Google icon, so users can log in with their Google credentials.

Follow these steps:

1. In the Authentication Methods area, click Add.

   The Select Authentication Methods screen displays a list of the authentication methods available in the tenant environment.

   Note: First, create authentication methods in the system before you perform this step. You define authentication methods through the User Console, using the Authentication Methods tasks. For more information, see Create Authentication Methods.

2. Select one or more authentication methods. The following types of authentication method are available:

   Basic

   Offers simple user name and password login.

   External IDP

   Offers log in through an external credential provider, such as Google or Facebook.

   Advanced Authentication

   Offers advanced authentication methods that have been configured for your environment, such as One Time Password (OTP) authentication.

   Note: Advanced Authentication methods only appear if you have purchased the Advanced Authentication Service.

   You can choose as many authentication methods, of any type, as you want. All the methods that you select are displayed on the login page that appears when a user attempts to access the application.

3. Click Select.

   The Create Application screen appears, updated with the list of authentication methods you selected.

4. (Optional) From the drop-down list, choose a default authentication method.

   Note: Advanced Authentication methods are never available as a default.

5. Configure Single Sign-On (see page 63).

(Optional) Configure Single Sign-On

Note: The option to configure single sign-on settings only appears in the User Console if you have purchased the SSO service.

During partnership configuration for an SSO application, a hosting administrator specifies a *federation attribute* for the partnership. The system uses this attribute to exchange information with the target software resource during single sign-on operations. For example, when configuring an SSO partnership between CA CloudMinder and salesforce.com, a hosting administrator chooses User ID as the federation attribute. The system retrieves this attribute from the database and forwards it in a SAML assertion to salesforce.com to facilitate single sign-on.

Some target software resources require the federation attribute to have a specific format. If this format differs from the format CA CloudMinder uses for the attribute, use the following steps to set the attribute value to the required format. This process is named setting the rule string for the attribute.

Note: Only configure the rule string if the software resource requires that the attribute take a format different from the way it is stored in the CA CloudMinder database.

Follow these steps:

1. In the Create Application screen, click Configure Single Sign On settings for the application.

   The Single Sign On configuration settings appear.

2. Select the Federation User Attribute.

   The attribute that you choose must match the assertion attribute that is indicated in the SSO partnership configuration for this application. If the attribute names do not match, users cannot successfully access this application through SSO. To confirm the assertion attribute name that is indicated in the partnership configuration, refer to your hosting administrator. SSO partnership configuration information is available in the CSP Console.

3. Configure the rule string for the Federation User Attribute.

   The rule string is the format that you want the attribute to take when the system passes it to the target software resource.

   Note: To learn the exact format that is required for this attribute, refer to your hosting administrator, or an administrator at the target software resource.

You have created an application and applied an authentication method. You have also configured single sign-on settings if applicable. You can now include this application in a service so that users can access the application.

Set Callback and Authentication URLs

Each client application you are configuring requires an OAuth-specific callback URL, and an authentication URL for your environment. These URLs are used to verify that the redirects performed during authentication are correct.

The OAuth Manager is a web utility that allows you to set callback and authentication URLs. It installs as part of the Layer 7 Gateway installation steps.

Note: Perform this procedure for each **OAuth Client** for which you configure CA CloudMinder as an OAuth authorization server.

Follow these steps:

1. Open OAuth Manager in a web browser:

   https://<GATEWAY1-HOST>:8443/<PREFIX>/oauth/manager

   Where GATEWAY1-HOST is the host name of Gateway one, and <PREFIX> is the Prefix you gave to the current tenant.

2. Log in with your Gateway administration credentials.
3. Click Manage Clients.
4. To add a new client, click Register Client.

   Note: To edit an existing client, select the client from the list, then click List Keys.

5. Select the Callback URL key and click Edit. Enter the following URLs:

   https://<CLUSTER-HOST>:8443/<PREFIX>/oauth/v2/client/authcode,
   https://<CLUSTER-HOST>:8443/<PREFIX>/oauth/v2/client/implicit

   Where <CLUSTER-HOST> is the VIP of the CA CloudMinder application-tier load balancer, and <PREFIX> is the prefix you selected for the current tenant.

   You set the prefix for your tenant during the Install OpenID Connect (see page 176) step.

6. Select the Environment key and click Edit. Enter the following authentication URL:

   https://<CLOUDMINDER-HOSTNAME>/chs/login/<TENANT-ID>/<APPLICATION-ID>/

   Where <CLOUDMINDER-HOSTNAME> is the hostname of the CA CloudMinder Administrative UI, <TENANT-ID> is the tenant tag specified during tenant creation, and <APPLICATION-ID> is the application tag specified during application creation.

   The Environment configuration sends the browser to the Administrative UI to make an attempt to access the CA CloudMinder Application associated with the OAuth client application.

You have now completed the steps necessary to configure an OAuth client to use CA CloudMinder as an OAuth authorization server. To configure additional OAuth clients, you repeat only the steps required once your tenant configuration is complete, indicated in the flow illustration (see page 167).

Chapter 6: Enable Domain Users to Access Applications Without Reauthenticating

Home realm detection enables users who have authenticated with their domain credentials to log into a target application without needing to select an identity provider on the CA CloudMinder login page.

For example, your company uses Google Apps, a software resource outside of your network environment. Users who have logged into the network with domain credentials should be able to access Google Apps without having to select an identity provider in the CA CloudMinder login page.

How Home Realm Detection Works

The following steps describe the process that takes place when home realm detection is enabled.

1. A user accesses a URL for an application that CA CloudMinder protects.

   The user is already logged into the corporate domain.

2. The proxy at the corporate site intercepts all requests that are directed to CA CloudMinder and injects the following header:

   ONPREM_AUTH_METHOD = *authentication method name*.

   authentication method name

   The name of the authentication method object in the User Console. The authentication method is associated with the application that the user is trying to access in the User Console.

3. CA CloudMinder receives the request with the header from the proxy and determines that the specified authentication method is associated with the particular application being accessed. CA CloudMinder redirects the user to the target application instead of the CA CloudMinder login page.

Enable Home Realm Detection

You enable home realm detection in the corporate proxy server.

Prerequisites:

- A proxy server that can intercept traffic to CA CloudMinder and insert a header into an HTTPS connection is installed and running.
- A partnership is configured between <stmdr>, installed at the corporate site, which acts as the external IdP and CA CloudMinder, which acts as the SP.
- CA CloudMinder protects the application that users want to access without having to reauthenticate. The application is defined in the User Console as follows:
  - The application definition is associated with an authentication method in the User Console.
  - The authentication method includes a reference to the URL that users are directed to.
- You have the exact name of the authentication method that is associated with the application definition in the User Console.

Configuration:

Configure the proxy to insert a header into all requests for CA CloudMinder.

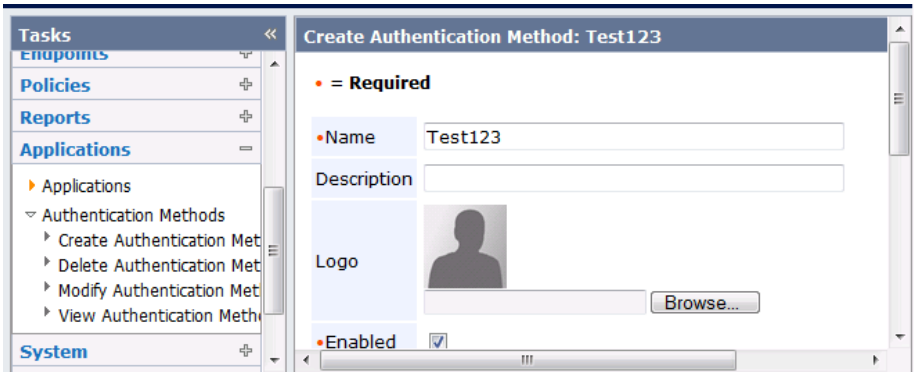
The header resembles the following example:

ONPREM_AUTH_METHOD = *authentication method name*.

authentication method name

The name of the authentication method object in the User Console.

In the following example, the authentication method object name is Test123.



The method for configuring the proxy server to insert a header depends on the proxy server that you are using. The following text illustrates a sample configuration for Fiddler, which is simulating proxy functionality:

```
if(oSession.HostnameIs("cloudMinder.domain.com")){
        oSesssion.oRequest["ONPREM_AUTH_METHOD"] = "Test123";
}
```

Note: For information on inserting headers into requests, see the documentation for your proxy server.

Example: How to Configure Home Realm Detection for Google Apps

The following example describes how to enable domain users at Forward, Inc. to access Google Apps without having to select an authentication method in the CA CloudMinder login screen.

This example assumes the following configuration:

- Corporate users are part of an Active Directory domain, and authenticate using Integrated Windows Authentication (IWA)
- CA SiteMinder is installed at the corporate site, and serves as the IdP in a SAML2 partnership. CA CloudMinder is the SP.

  In CA SiteMinder, the following configuration is defined:

  - IWA authentication is enabled.
  - The following resource is protected by IWA authentication:

    /affwebservices/redirectjsp/redirect.jsp

    - The resource URL above is specified as the Authentication URL in the partnership, as follows:

      https://test.forwardinc.com/affwebservices/redirectjsp/redirect.jsp
- CA CloudMinder also acts as the IdP in a federated partnership with Google Apps (the SP).

  In this partnership, the following configuration exists:

  - The Delegated Administration URL is the URL for the target application.

    https://*domain*/chs/login/*tenant_name*/*application in User Console*

    For example:

    https://test2.cloud.com/chs/login/ForwardInc/GoogleApps
  - The SSO Service URL for the partnership resembles the following:

    https://*domain*/affwebservces/public/saml2sso?SPID=*SPID of target application*

    For example:

    https://test2.cloud.com/affwebservces/public/saml2sso?SPID=google.com/a/ggl.test.com

    This URL must match the Launch URL that you define in the application (see page 193).
- A proxy server that can intercept traffic to CA CloudMinder and insert a header into an HTTPS connection is installed and running at the corporate site.

Follow these steps:

1. Create authentication method (see page 192).
2. Create an application (see page 193).
3. Configure the on-premise proxy (see page 193).

Configure the Authentication Method

Once you define the authentication method, you associate it with an application in the tenant User Console.

Follow these steps:

1. Log in to the User Console.
2. Navigate to Applications, Authentication Methods, Create an Authentication method.
3. Specify a name for the authentication method.

   The name you specify is added to the ONPREM_AUTH_METHOD header that the on-premise proxy sends to the credential handling service to determine if the user is prompted for credentials.

   For this example, specify the following authentication method name:

   Test123Auth
4. Select External IDP as the authentication method.
5. Specify the following values in the External IDP Type Configuration section:

   Authentication Method Scheme

   Select Other.

   Authentication URL

   Must match the SSO Service URL in the CA SiteMinder (IdP) to CA CloudMinder (SP) partnership. Use the following format:

   https://*domain*/affwebservices/public/saml2sso?SPID=*SPID of partnership*
6. Click Submit.

Configure the Application

You create an application to define how users access a software resource. For example, when you configure an application, you define what type and level of security protects the resource.

Follow these steps:

1. Log in to the User Console with an account that has application management privileges.

   For example, the default Tenant Administrator role has the appropriate privileges.

2. From the navigation menu, select Applications, Create Application.
3. Enter a name and description.

   For this example, specify TestApp as the name.

4. In the Launch URL field, enter the IdP-initiated partnership URL for the target application as follows:.

   https://cloud1.test.com/affwebservces/public/saml2sso?SPID=*SPID of target application*

   For this example, enter the following URL:

   https://test2.cloud.com/affwebservces/public/saml2sso?SPID=google.com/a/ggl.test.com

5. Add the authentication method (see page 192).

   The authentication you created in the previous step is called Test123Auth.

6. Click Submit.

Configure the Proxy

Configure the on-premise proxy to forward a header to the Credential Handler application.The header must be named ONPREM_AUTH_METHOD and contain the name of the authentication method for the target application, as defined in the tenant User Console. In this example, the header resembles the following:

ONPREM_AUTH_METHOD=Test123Auth

Troubleshooting Home Realm Detection

If issues occur, check the /opt/CA/secure-proxy/proxy-engine/logs/chsLogin.log file. Search for the following string:

"JSON parsing done, found default auth method; redirecting to *URL*"

Test the string by copying the URL into a browser window to see if the federated partnership works correctly.

If the string does not exist in the log file and the user is directed to the login page, verify the following configuration settings:

- The partnership uses the SAML 2.0 protocol
- The ONPREM_AUTH_METHOD header in the proxy specifies the name of the authentication method for the application exactly as it is specified in the User Console.
- The application definition in the User Console correctly specifies the authentication method in the ONPREM_AUTH_METHOD header.

Chapter 7: How to Set Up the Security Token Service

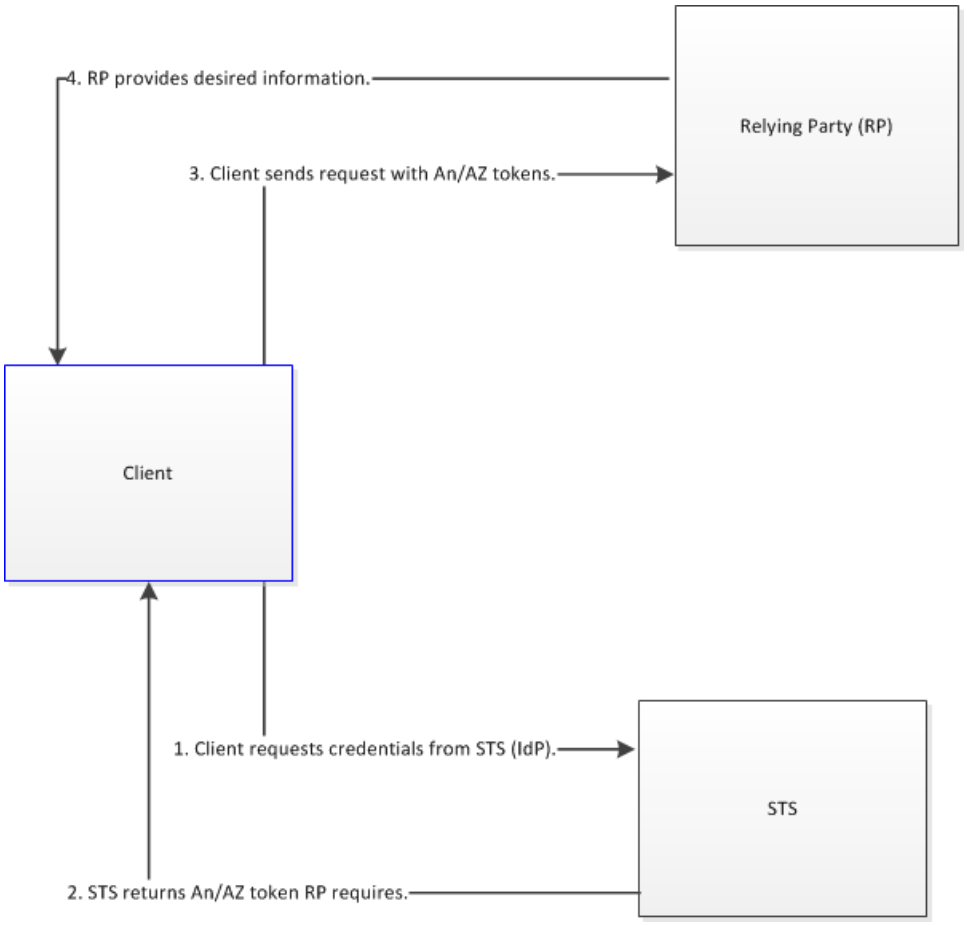
Overview of STS Set Up

The Security Token Service (STS) provides a WS-Trust-based mechanism for token issuance and translation. The WS-Trust specification includes extensions to the WS-Security standard. WS-Trust specifies the following:

- A general framework for token exchanges
- A model for brokering trust in web services
- A Security Token Service framework
- Protocols for issuing security tokens

The main function of an STS is to serve as a third party that can provide credentials for a relying party. The STS is an authority trusted by the client and the relying party. The STS client application is WS-Trust literate; it generates token requests. You can use the STS to issue a security token, which is a collection of claims such as name, role, and authorization code required for the client to access the relying party.

The following diagram illustrates authentication to a relying party using the STS:



Each instance of the STS contains an embedded SOA agent. Web service requests for security tokens are authenticated and authorized with the same functionality as the CA SiteMinder SOA product. The STS supports WS-Trust 1.3/1.4 compliant requests for the issuing of a variety of security tokens.

The following list details the authentication schemes and token types that the STS supports for each:

XML DCC

WS-Username (digest), WS-X509, WS-SAML Holder-of-key (SAML 1.1 & 2.0), WS-SAML Bearer (SAML 2.0), SMSession

XML DSig

WS-X509, WS-SAML Holder-of-key (SAML 1.1 & 2.0), WS-SAML Bearer (SAML 2.0), SMSession

WS-Username (plain and Digest)

WS-Username (digest), WS-X509, WS-SAML Holder-of-key (SAML 1.1 & 2.0), WS-SAML Bearer (SAML 2.0), SMSession

WS-X509

WS-SAML Holder-of-key (SAML 1.1 & 2.0), WS-SAML Bearer (SAML 2.0), SMSession

WS-SAML Bearer (SAML 2.0)

WS-SAML Holder-of-key (SAML 1.1 & 2.0), WS-SAML Bearer (SAML 2.0), SMSession

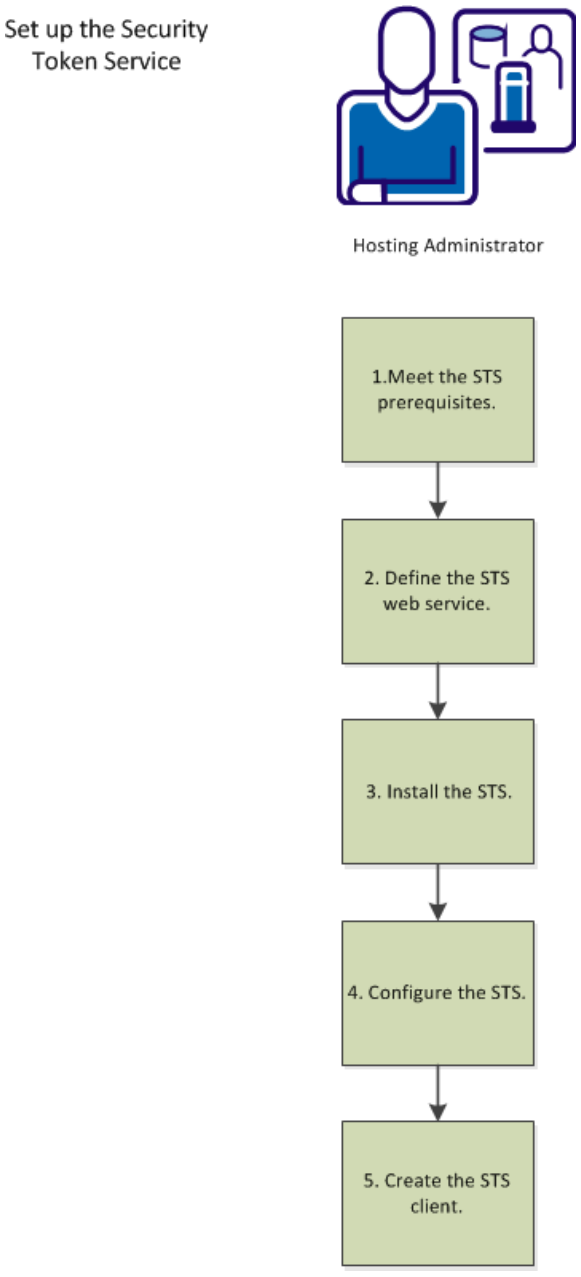
WS-SAML Holder-of-key (SAML 1.1 & 2.0)

WS-SAML Holder-of-key (SAML 1.1 & 2.0), WS-SAML Bearer (SAML 2.0), SMSession

SMSession

WS-Username (digest), WS-X509, WS-SAML Holder-of-key (SAML 1.1 & 2.0), WS-SAML Bearer (SAML 2.0), SMSession

The following diagram illustrates the process of setting up the STS:



1. Meet the STS prerequisites (see page 199).
2. Define the STS web service (see page 199).
3. Install the STS web service (see page 201).
4. Configure the STS (see page 201).
5. Create the STS Client (see page 202).

Meet the STS Prerequisites

Before you create an STS web service, verify that the STS host system meets the prerequisites.

Follow these steps:

1. Install the unlimited crypto policy.jar files from the Oracle web site. Download the files into the $jre/lib/security folder.
2. Install JBoss 5.1 EAP to run with the previously installed unlimited crypto policy JDK/JRE.

Define the STS Web Service

After you have verified that all the prerequisites are met, you can create the STS web service. The steps that are listed following are the minimum required.

Follow these steps:

1. Log in to the CSP Console.
2. Navigate to Web Services, Create STS Web Service.
3. Enter the name and an optional description in the General section.
4. Click the Add/Remove button in the User Directories section.
5. Verify that the user directories you are interested in appear in the Selected Members panel.
6. Click OK.
7. Click the Add button in the Authentication and Token Generation section

8. Enter a name for the end point and an optional description in the General section. The description appears in the WSDL file and best includes information about the SOA authentication scheme.
9. Select a SOA authentication scheme from the list. You are responsible for defining one or more SOA authentication schemes.
10. Select a response for one or more of the supported token types. You are responsible for defining the responses. You define the responses in the Policies, Global, Global Responses dialog.
11. Click OK.
12. Enter optional specifications for the Web Service Definition, Relying Party, and Session sections.

    Note: If you plan to sign RSTR responses (the default behavior), be sure that you choose a signing certificate in the STS UI. Alternatively, you can add a private key/certificate combination named defaultenterpriseprivatekey to the keystore.
13. Click Submit.

The STS web service is ready for installation and configuration.

Install the STS

The STS installer mirrors the other service installers in functionality. It creates a central directory (user specified) that the web service has access to, for example, /opt/sts. Under the /sts directory, the installer creates two subdirectories: /config and /log.

The installation executable is ca-sts-1.1-rhel30.bin, which is included in the CAM-SMPS kit. Note that this version of CA CloudMinder only supports the RedHat Linux 64-bit operating system.

Follow the InstallAnywhere instructions. When the wizard completes the installation, you can find the following files in the /config directory:

- JavaAgent.conf
- agent-log4j.xml
- JSAMLAssertionStrings.properties
- JSAMLProtocolStrings.properties
- XmlToolkit.properties
- urlToResourceMap.xml
- urlToResourceMap.xsd
- soap11.xsd
- soap12.xsd

Before you run the STS configuration program, perform these two steps:

1. Move the cryptoj.jar file from *<sts_home>/<sts_instance_name>*/sts/lib to $jre/jre/ext.
2. Set up the JSafeJCE as a crypto provider.

   Note: Instructions for configuring the JVM to use the JSafeJCE Security Provider are located in all of the CA SOA Agent guides.

Configure the STS

The STS configuration command is ca-sts-config.sh. The configuration wizard prompts you for the following information:

- The name of the web service that you specified in the UI.
- The location of a JDK with unlimited crypto strength
- Host registration information
- The root directory of the JBoss application server

The configuration adds the SmHost.config file to the installation directory.

As a workaround, be sure to set the WSDMResourceIdentification parameter to No in the $sts/config/XmlToolkit.properties file. Otherwise, the authorization can fail because of a resource identification check.

After you complete the configuration, you can start an STS client application at any time.

Important! The STS service locates its configuration using an Agent Config Object (ACO). The ACO has the same name as the service. When you rename the service, you also rename the ACO, In this case, be sure to edit JavaAgent.conf in their STS install directory with the updated ACO name

Create the STS Client

The function of the STS client is to issue compliant requests for a variety of security tokens. The client has to be WS-Trust literate. A WSDL file describes the web service interface. You can find the WSDL file from the the base URL of the service, as in the following example: http://hostname:80/STS?wsdl. You can run the WSDL file through your choice of code generation tools to generate the foundational code for the STS client.

The STS supports WS-Trust Soap requests (RST) and responses (RSTR). The RST specifies one or more token types. The STS supports the following values for <Token Type>:

http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-username-token-profile-1.0

Specifies the WS-Security Username token.

http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-x509-token-profile-1.0#X509v3

Specifies the WS-Security X509v3 token.

http://docs.oasis-open.org/wss/oasis-wss-saml-token-profile-1.1#SAMLV1.1

Specifies the WS-Security SAMLv1.1 Assertion with Holder-of-Key confirmation method.

http://docs.oasis-open.org/wss/oasis-wss-saml-token-profile-1.1#SAMLV2.0

Specifies the WS-Security SAMLv2.0 Assertion with Holder-of-Key confirmation method.

urn:oasis:names:tc:SAML:2.0:cm:bearer

Specifies the WS-Security SAMLv2.0 Assertion with Bearer confirmation method.

http://www.ca.com/siteminder/smsession

Specifies the proprietary SMSESSION token (XML format).

The RST can also include an <AppliesTo> element, which specifies the relying party that will consume the token. The STS is configured with a set of known relying parties – including the Response required to generate the token needed by the relying party.

The <AppliesTo> value can be a simple URL, as in the following example:

```
<AppliesTo
xmlns="http://schemas.xmlsoap.org/ws/2004/09/policy">http://some-relying-party.customer.com</AppliesTo>
```

The <AppliesTo> value can also be a WS-Addressing element:

```
<wsp:AppliesTo xmlns:wsp="http://schemas.xmlsoap.org/ws/2004/09/policy"
            xmlns:wsa="http://www.w3.org/2005/08/addressing">
  <wsa:EndpointReference>
    <wsa:Address>http://some-relying-party.customer.com</wsa:Address>
  </wsa:EndpointReference>
</wsp:AppliesTo>
```

If both a <TokenType> and an <AppliesTo> value are present, the <AppliesTo> value takes precedence.