CA SiteMinder® Federation

Legacy Federation Guide

12.52 SP1

CA Technologies Product References

This document references the following CA Technologies products:

- CA SiteMinder®
- CA SiteMinder® Web Agent Option Pack
- CA SiteMinder® SPS

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.

Documentation Changes

No updates have been made to the 12.52 SP1 documentation as a result of issues found in previous releases.

The following updates have been made to the 12.52 documentation, as a result of issues found in previous releases.

- Removed all references to the SAML Affiliate Agent throughout the guide. This product is no longer supported.
- Command Options for smfedexport (see page 341)—Added the -decryptionkeyalias command option for the smfedexport utility.

Contents

Chapter 15: Configure a SAML 2.0 Service Provider 231

Chapter 23: Federation Web Services URLs Used by SiteMinder 367

Chapter 24: Troubleshooting Legacy Federation 385

Chapter 1: Legacy Federation Introduction

This section contains the following topics:

Terminology for Partners in a Federation

This guide uses the terms *asserting party* and *relying party* to identify sides of a federated relationship.

The party that generates assertions is referred to as the asserting party. The asserting party can be:

- SAML 1.x producer
- SAML 2.0 Identity Provider
- WS-Federation Account Partner

The party that consumes assertions for authentication purposes is referred to as the relying party. The relying party can be:

- SAML 1.x consumer
- SAML 2.0 Service Provider
- WS-Federation Resource Partner

A site can be act as an asserting party (producer/IdP/AP) and a relying party (consumer/SP/RP).

Components for Legacy Federation

Legacy federation is based on configuring CA SiteMinder® objects, such as affiliate domains, affiliate partners, authentication schemes, and policies to protect federated resources.

For more information about CA SiteMinder® federation offerings, legacy federation use cases, and process flow diagrams, see *Federation in Your Enterprise*.

Legacy federation uses several components:

SAML Assertion Generator

A Policy Server component that creates SAML assertions at the asserting party.

The SAML assertion generator creates an assertion for a user who has a session at a producer/IdP site. When a partner requests a SAML assertion, the Web Agent invokes the SAML assertion generator. The assertion generator creates an assertion that is based on a user session and information in the policy store.

The assertion generator processes the assertion according to the authentication profile or binding configured, as follows:

- SAML artifact profile/binding

  The assertion generator stores the assertion in the CA SiteMinder® session store. A reference to the assertion is returned to the Web Agent in the form of a SAML artifact.

- SAML POST profile/binding

  CA SiteMinder® returns the assertion by way of a browser as a SAML response embedded in an HTTP form.

The Web Agent is responsible for sending the SAML artifact, SAML response, or WS-Federation security token response to the relying party in accordance with the SAML profile. At the relying party, a client must be available to process the SAML artifact or response message. If CA SiteMinder® is the relying party, the client can be the SAML 1.x credential collector or the SAML 2.0 assertion consumer.

You can customize the content of a SAML assertion by configuring the assertion generator plug-in. This plug-in lets you customize the content for your federated environment.

WS-Federation Assertion Generator

A Policy Server component that creates WS-Federation RequestSecurityTokenResponse messages containing SAML assertions.

The WS-Federation assertion generator creates a SAML 1.1 assertion for a user who has a session at an Account Partner. When a user requests a resource, the Web Agent invokes the WS-Federation assertion generator at the Policy Server. The Policy Server creates an assertion that is based on the user session and information configured in the policy store. The assertion generator then places the assertion in a WS-Federation RequestSecurityTokenResponse message.

The Web Agent sends the security token response message, through a browser, to the relying party in accordance with the WS-Federation Passive Requestor profile. At the Resource Partner, a client, such as WS-Federation Assertion Consumer must be available to process the assertion.

You can customize the content of the SAML assertion by configuring the assertion generator plug-in. This plug-in lets you customize the content for your federated environment.

Federation Authentication Schemes

A Policy Server component that validates SAML or WS-Federation assertions and maps assertion data to a local user at the relying party.

The supported authentication schemes are:

- SAML 1.x artifact
- SAML 1.x POST
- SAML 2.0 (artifact and POST binding)
- WS-Federation

Federation Web Services

A Web Agent component that supports assertion retrieval, session synchronization and notification alerts at the asserting party. At the relying party, these services collect assertions.

Legacy Federation Authentication Schemes

Legacy federation supports the following authentication schemes:

- SAML 1.x artifact
- SAML 1.x POST
- SAML 2.0
- WS-Federation

Each authentication scheme enables CA SiteMinder® at the relying party to process SAML assertions. Upon receiving an assertion, the authentication scheme

- Validates the SAML assertion
- Maps assertion data to a local user
- Establishes a CA SiteMinder® session at the site consuming the assertion

The SAML authentication scheme is where you map remote users at the asserting party to local users at the relying party. User mapping enables the authentication scheme to locate the correct user record for authentication.

Federated Single Sign-on with Security Zones

A CA SiteMinder® environment can be set up to include a web application environment for web service protection and a federation environment for federated resource protection. This method can make a CA SiteMinder® deployment more efficient.

Certain federation features require a persistent user session, which means that the SAML assertion is stored in the session store of the Policy Server.

These features include:

Artifact Single sign-on

For SAML 1.x and SAML 2.0, the SAML assertion is stored in a persistent session that the relying party retrieves later.

Single Logout

(SAML 2.0 Single Logout and WS-Fed Signout) at producer and consumer sites. Partner data is stored in a persistent user session to facilitate notification of partners during a federated logout.

Use of persistent user sessions slow down performance because of the required calls to the session store to retrieve assertions or handle log-out requests. To limit the performance impact, use security zones.

A security zone is a segment of a single cookie domain. The security zone lets you partition applications to permit different security requirements for resource access. All applications in a single zone permit single sign-on to one another. If an application is in another zone, the trust relationship that you configure determines single sign-on.

For federated applications at the asserting party, implement the following setup:

- Create a dedicated security zone.
- Create a different zone for all non-federated applications.
- Configure the federated zone to trust the non-federated zone.

The use of different zones confines calls to the session server for only federated applications.

Note: In a federated environment, you can only configure Web Agents to use security zones. Secure Proxy Agents and Application Server Agents do not support this feature.

To configure security zones, enter values for the following Web Agent parameters:

SSOZoneName

Identifies a single sign-on security zone. The zone name is added to the cookie domain name to associate the zone with the domain.

Note: This item supports only English-language characters. Characters from other languages are not supported.

SSOTrustedZone

Displays an ordered list of trusted security zones. Defining zones and trusted zone lists determine the cookies that the Web Agent is able to read and write.

These parameters are part of an Agent Configuration Object or a local Agent configuration file.

For more information about security zones, see the *Web Agent Configuration Guide.*

Secure Proxy Server Federation Gateway

The CA SiteMinder® SPS federation gateway offers a proxy-based solution to access control in a federated network. A traditional proxy typically serves a group of users requesting internet resources. The SPS federation gateway is a reverse proxy, meaning it acts on behalf of users requesting resources from an enterprise.

The SPS federation gateway is a self-contained system. The gateway has its own servlet engine and web server built-in. The SPS gateway relies on its proxy engine to handle requests from federated partners to protected resources. Enhancing SPS to work as a federation gateway allows quick deployments.

As a component of legacy federation, the SPS federation gateway can replace the Web Agent and Web Agent Option Pack to provide the services of the Federation Web Services application. A single SPS federation gateway can limit the amount of configuration that is required for access to resources by limiting the need for many Web Agents.

Note: CA SiteMinder® SPS is a separately licensed product from CA SiteMinder®.

Internationalization in Legacy Federation

Legacy federation supports the following features for I18N internationalization:

- Legacy federation configuration objects, Java and C++ code are encoded in UTF-8 format for the internationalization purposes.
- The creation and the consumption of default and customized assertions with multibyte user IDs and attribute values.
- Encoded target and redirect URLs. These URLs are encoded per HTTP 1.1 RFC 2616 so multibyte path and file names are handled correctly.

If assertions contain multibyte characters, set the LANG setting of your operating system to the following UTF-8 format:

LANG=xx_xx.UTF-8

For example, for Japanese, the entry would be:

LANG=ja_JP.UTF-8

Debugging Features

Legacy federation components log specific events to monitor and debug activity across the federated network.

Web Agent log

Displays information about any request to generate a SAML assertion at a producer site.

Federation Web Services log

Displays information about requests to retrieve SAML assertions and to consume SAML assertions.

Policy Server log

Displays the results of calls from the SAML assertion generator and SAML artifact authentication scheme. Also displays Policy Server trace messages that you configure using the Policy Server Management Profiler or using one of the provided profiler template files.

Web Agent Option Pack logs

Displays FWS trace messages that you configure using the FWSTrace.conf file or using one of the provided trace template files.

APIs for Legacy Federation

The following APIs provide support for legacy federation.

- Policy Management API
- Java Message Consumer Plugin API
- Java Assertion Generator Plugin API

Policy Management API

The C and Perl Policy Management APIs provide new language elements in support of CA SiteMinder® federation. These new language elements include:

- C structures and Perl packages for federation objects. The objects include affiliate domains, Affiliates, Identity and Service providers, Resource and Account Partners.
- C functions and Perl methods for SAML 1.x, SAML 2.0, and WS-Federation configuration.

- SAML 2.0 metadata constants.
- WS-Federation metadata constants.

For more information about the Policy Management API, see *the CA SiteMinder® Programming Guide for Perl* or the *CA SiteMinder® Programming Guide for C*.

Java Message Consumer Plugin API

The CA SiteMinder® Java MessageConsumerPlugin API implements the SAML 1.x, SAML 2.0 and WS-Federation Message Consumer Extension interface. This API allows you to perform your own processing for user disambiguation and authentication. After you customize code for your own requirements, integrate the custom plug-in into CA SiteMinder® to further process and manipulate a SAML assertion response or the WS-Federation security token response.

For more information, see the *CA SiteMinder® Programming Guide for Java*.

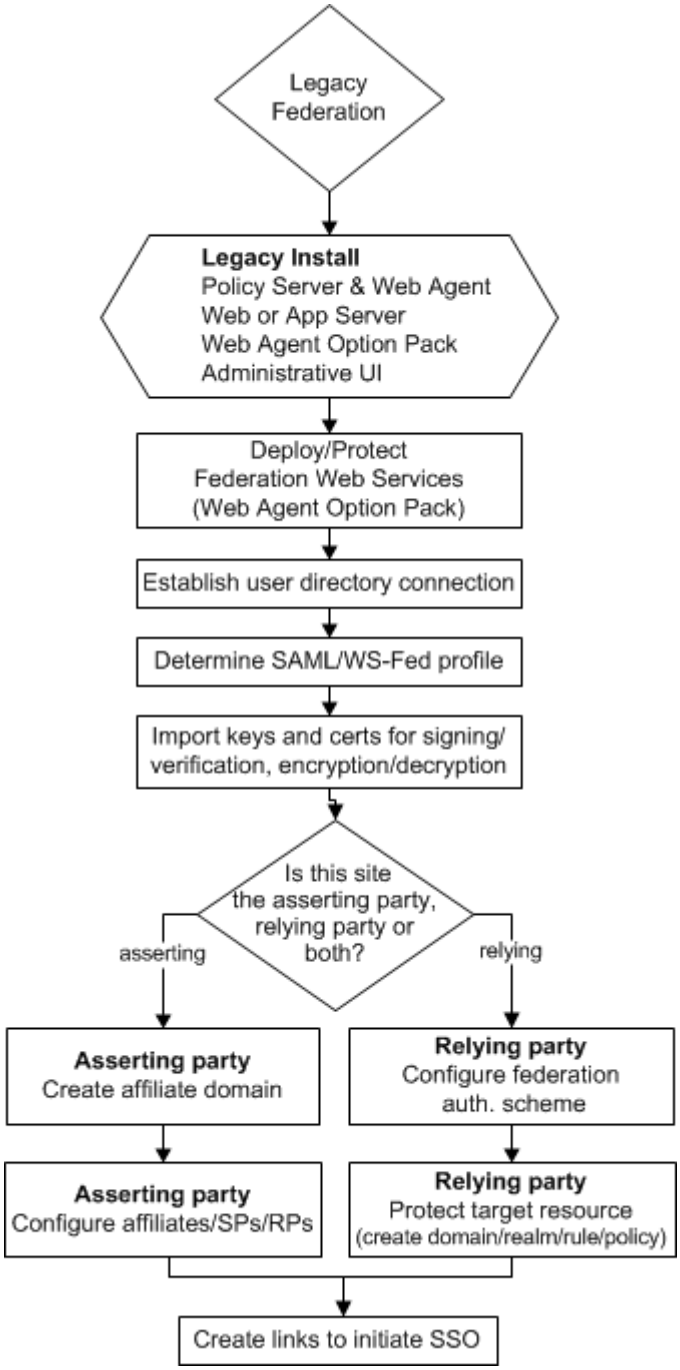
Java Assertion Generator Plugin API

The SiteMinder Java Assertion Generator Plugin API implements the Assertion Generator Framework. Using the plug-in, you can modify the assertion content for your business agreements between partners and vendors.

For more information, see the *CA SiteMinder® Programming Guide for Java*.

Legacy Federation Configuration Flow Chart

The following flow chart highlights the general process for configuring legacy federation.



Chapter 2: Use a Sample Configuration to Learn About Legacy Federation

This section contains the following topics:

Manual CA SiteMinder®-to-CA SiteMinder® Deployment Overview

You can accomplish a deployment manually. The manual deployment tasks begin with a simple configuration, single sign-on with POST binding. By starting with a basic configuration, you can complete the least number of steps to see how federation works.

After getting POST single sign-on to work, additional tasks, such as configuring the artifact binding, digital signing, and encryption are described. You can add these features to reflect a real production environment.

Important! The deployment exercise is only for SAML 2.0. These procedures do not apply to a SAML 1.x or WS-Federation configuration.

The manual deployment examples are different from the sample application deployment in the following ways:

- The deployment that is described is set up across two systems, with a Policy Server and Web Agent on each system. The two systems represent the IdP and the SP.
- Additional features are documented for the manual configuration that the sample application does not set up, including:
  - Configuring SSL for the artifact back channel
  - Adding an attribute to an assertion

- Digitally signing and verifying an assertion
- Encrypting and decrypting an assertion

The procedures throughout the manual deployment use sample data. To use data from your environment, specify entries for your Identity Provider and Service Provider configuration.

Confirm that Required Components are Installed

Federation requires that the following components are installed:

- CA SiteMinder® Policy Server
- Administrative UI
- Web Agent
- Web Agent Option Pack

This sample federation deployment example assumes that these components are installed and working.

Optionally, set up these features:

- Enable a web or application server for SSL communication.

  SSL is optional for securing communication across the back channel for HTTP-artfact single sign-on.

- Set up a database to be enabled as a session store.

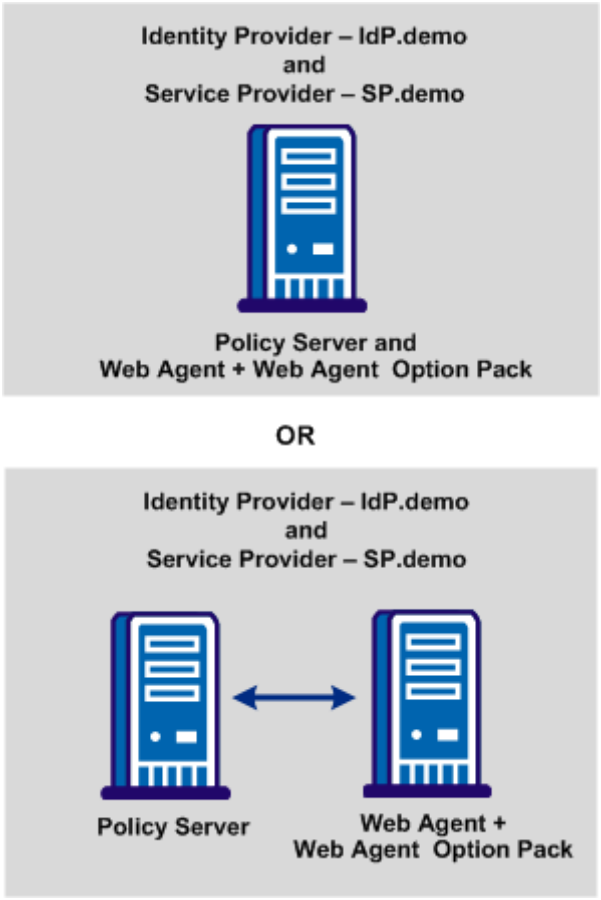
  The session store is required for storing the assertion at the asserting party when using HTTP-Artifact binding. The session store is also required for single logout at either party.

  For instructions on setting up a session store database, see the *Policy Server Installation Guide*. Enable the session store using the Policy Server Management Console. See the *Policy Server Administration Guide for instructions*.

Sample Federation Network

The sample websites in the CA SiteMinder® federated network are an Identity Provider named idp.demo, and a Service Provider named sp.demo. A business partnership is established between idp.demo and sp.demo.

The following illustration shows the sample federated network.



Identity Provider Data for a Basic Configuration

IdP.demo is the Identity Provider. The following table contains sample data for the most basic SAML 2.0 POST single sign-on configuration.

Identity Provider Component	Sample Network
IdP Policy Server	**Server**: www.idp.demo:80 **Server type**: IIS Web Server

Identity Provider Component	Sample Network
IdP policy store	**IP Address**: www.idp.demo:389 **Storage**: LDAP (Sun One Directory Server) **Root DN**: o=idp.demo **Admin Username**: cn=Directory Manager **Password**: federation
User store	**Directory name:** IdP LDAP **Server**: www.idp.demo:42088 **Server Type**: Sun One Directory Server (LDAP) **User store**: The LDAP directory contains the following users: ■ user1 ■ user2 **userpassword**: test **mail**: <*user_name*>@idp.demo **Root**: dc=idp,dc=demo **Start**: uid= **End**: ,ou=People,dc=idp,dc=demo
IdP Web Agent with Web Agent Option Pack	**Server**: www.idp.demo:80 **Server Type**: IIS Web Server **Agent name**: idp-webagent
Assertion Consumer Service URL	**URL**: http://www.sp.demo:81/affwebservices/ public/saml2assertionconsumer
Assertion Retrieval Service URL	**URL:** http://www.idp.demo:80/affwebservices/assertionretriever
Authentication URL	**URL**: http://www.idp.demo/siteminderagent/ redirectjsp/redirect.jsp

Identity Provider Data for an Advanced Configuration

The following table contains sample data for more advanced SAML 2.0 features, such as the artifact profile, signing and encrypting assertions.

Identity Provider Component	Sample Network
Session store	**Server:** www.idp.demo **Database type:** ODBC **Database Source Information:** SiteMinder Session Data Source **User Name:** admin **Password**: dbpassword
SSL-enabled server	**Server:** www.idp.demo:443 **Server Type**: IIS 6.0 Web The web server with the Web Agent Option Pack is SSL-enabled for artifact binding
Certificate of the Certificate Authority (CA)	**Certificate of CA**: docCA.crt **DER-encoded Cert:** docCA.der This CA signs the server-side certificate to enable SSL
Private key/certificate pair to sign SAML responses	**Certificate**: post-cert.crt **Private key**: post-pkey.der **Password**: fedsvcs
Certificate (public key) for encryption	**Public key**: sp-encrypt.crt
Attribute to include in assertion	**Attribute**: unspecified (default) **Attribute Kind**: User DN **Variable Name**: firstname **Variable Value**: givenname

Service Provider Data for a Basic Configuration

The Service Provider is SP.demo. The following table contains sample data for the most basic SAML 2.0 POST single sign-on configuration.

Service Provider Component	Sample Network
SP Policy Server	**Server**: www.sp.demo:80 **Server type**: IIS Web Server
SP policy store	**IP Address**: www.sp.demo:389 **Storage**: LDAP (Sun One Directory Server) **Root DN**: o=ca.com **Admin Username**: cn=Directory Manager **Password**: federation
User Store	**Directory name:** SP LDAP **Server**: www.sp.demo:32941 **Server Type**: LDAP (Sun One Directory Server) **User store**: The LDAP directory contains the following users: ■ user1 ■ user2 **userpassword**: customer **mail**: <user_name>@sp.demo **Root**: dc=sp,dc=demo **Start**: uid= **End**: ,ou=People,dc=sp,dc=demo
SP Web Agent and Web Agent Option Pack	**Server**: www.sp.demo:81 **Server type**: Sun ONE 6.1 Web Server **Agent name**: sp-webagent
Single Sign-on Service	**SSO Service**: http://www.idp.demo:80/affwebservices/public/saml2sso
Target Resource	**Target Resource**: http://www.sp.demo:81/ spsample/protected/target.jsp

Service Provider Data for an Advanced Configuration

The following table lists sample data for more advanced SAML 2.0 features, such as setting up the artifact profile, signing and encrypting assertions.

Service Provider Component	Sample Network
Artifact Resolution Service	**Resolution Service**: https://www.idp.demo:443/affwebservices/saml2artifactresolution
Certificate of Certificate Authority	**Certificate of CA**: docCA.crt **DER-encoded cert**: docCA.der This CA signs the server-side certificate to enable SSL
Certificate (public key) Used to verify signature of SAML responses	**Certificate**: post-cert.crt
Private key/certificate pair Used for decryption and digital signing	**Private key**: sp-encrypt.der **Public key**: sp-encrypt.crt **Password**: fedsvcs **Issuer DN**: CN=Certificate Manager,OU=IAM,O=CA.COM **Serial Number**: 008D 8B6A D18C 46D8 5B

Set Up the Identity Provider for the Sample Network

To deploy legacy federation at the Identity Provider, the following sections detail the tasks. The entries in each section reflect the sample data that is provided for a basic configuration.

Note: These procedures assume that you have already installed the required components.

Set Up the IdP User Store

At the Identity Provider, a user store with users defined is required. The Identity Provider can create assertions for these users. In this deployment, the user store is a Sun ONE LDAP user directory. The Sun ONE Server Console is used to add users to this user store.

To configure the user store

1. Add the following users:
   - user1
   - user2
2. Fill in the attributes for user1 and user2 as follows:

   user1

   userpassword: test

   mail: user1@idp.demo

   user2

   userpassword: test

   mail: user2@idp.demo

   Important! The email address must be the same in the Service Provider user store for the same users.
3. Enable trace logging (see page 35).

Point the Policy Server to the IdP LDAP Policy Store

In this deployment, an LDAP policy store is used. Verify that the Policy Server is pointing to the LDAP policy store.

Note: The procedure assumes that you know how to add users to the user store in your deployment.

Follow these steps:

1. Open the Policy Server Management Console.
2. Select the Data tab.
3. Complete the following fields:

   Databases

   Policy Store

   Storage

   LDAP

   IP Address (LDAP directory)

   www.idp.demo:389

   Root DN

   o=idp.demo

Admin Username

cn=Directory Manager

Password

password

Confirm Password

password

4. Click OK to save your changes and exit the console.
5. Go to Set Up the IdP User Store (see page 33).

Enable Policy Server Trace Logging at the IdP

At the Identity Provider, enable logging for the Policy Server. You can view the log file smtracedefault.log to examine trace messages about single sign-on and single log out. This log file is in the directory *policy_server_home*/siteminder/log.

Follow these steps:

1. Open the Policy Server Management Console.
2. Click on the Profiler tab and customize the contents of the trace log.

   Note: Include the Fed_Server component in the log to see the federation trace messages.

   You configure trace logging at the Policy Server using the Policy Server Management Console.
3. Install the IdP Web Agent.

Configure the Web Server with the Web Agent Option Pack

Configure the Federation Web Services (FWS) application for the sample deployment.

To set up FWS:

- Install the JDK for Federation Web Services (see page 36)
- Install and Configure ServletExec to work with FWS at the IdP (see page 36)
- Configure the AffWebServices.properties File at the IdP (see page 38)
- Test Federation Web Services at the IdP (see page 39)

Install the JDK for Federation Web Services

The Web Agent Option Pack requires a JDK to run the Federation Web Services application.

For the correct JDK version, go to the Technical Support site and search for the CA SiteMinder® Platform Support Matrix for the release.

Install and Configure ServletExec to work with FWS at the IdP

For FWS to operate, you can install ServletExec or any supported application server. This sample network uses ServletExec on an IIS 6.0 Web Server.

Note: CA SiteMinder® 12.52 SP1 is shipped with a ServletExec license key file named ServletExec_AS_6_license_key.txt. If you do not have this license key, contact CA Technical Support. From this license file, copy the license key and enter it in the ServletExec License dialog of the ServletExec Administration Console. For instructions on licensing ServletExec, see ServletExec documentation, available at the New Atlanta Communication http://www.newatlanta.com website.

Be sure to apply the most current hot fixes for the supported version of ServletExec you are using. The hot fixes are necessary for Federation Web Services to work with ServletExec. To obtain hot fixes, go to the website for New Atlanta Communication.

To set up ServletExec

1. Install ServletExec. For more information, see the New Atlanta documentation.
2. Open the ServletExec Administration Console.
3. Under Web Applications, select manage.

   The Manage Web Applications dialog opens.
4. Click Add a Web Application.
5. Enter the following information:

   Application Name

   affwebservices

   URL Context Path

   /affwebservices/

   Location

   C:\program files\ca\webagent\affwebservices

   Note: The location of affwebservices in your setup can be different. Enter the correct location.
6. Click Submit.

7. Exit the ServletExec Console.
8. Modify the directory security settings for the IIS default user account.

Important! The IIS user account must have proper rights for IIS to allow any plug-in to write to a file system. Therefore, for Federation Web Services to work with ServletExec, modify the directory security settings for the IIS default user account.

More Information:

Enable ServletExec to Write to the IIS File System (see page 37)
Configure the FWS Properties File at the IdP (see page 38)

Enable ServletExec to Write to the IIS File System

The IIS server user account must have proper rights for IIS to allow a plug-in to write to its file system. For ServletExec to write to the federation log files, the anonymous user account that is associated with ServletExec must have permissions to write to the file system.

Follow these steps:

1. Open the IIS Internet Information Services Manager on the system where ServletExec is installed.
2. Navigate to Web Sites, Default Web Site.

   The set of applications is displayed in the right pane.
3. Select ServletExec and right-click Properties.
4. Select the Directory Security tab in the Properties dialog.
5. Click Edit in the Authentication and access control section.

   The Authentication Methods dialog opens.
6. Set the controls as follows.
   a. Select Enable Anonymous Access.

      For anonymous access, enter a name and password of a user account that has the permissions to right to the Windows file system. To grant this right to a user account, see Windows documentation. For example, you can use the IUSR Internet Guest account for anonymous access.

   b. Clear Basic authentication.

   c. Clear Integrated Windows authentication.

7. If prompted, apply the security changes to all child components of the web server.
8. Restart the web server.

The user account that is associated with ServletExec can now write to the IIS file system.

Follow these steps:

1. Open Control Panel, Administrative Tools, Local Security Policy, Local Policies, User Rights Assignment.

   The Local Security Settings dialog displays.
2. Double-click Act as part of the operating system.

   The Act as part of the operating system Properties dialog opens.
3. Add the anonymous user account to the Local Security Setting dialog.
4. Click OK.
5. Exit from the control panel.
6. Optionally, we strongly recommend that you look at the Agent Configuration Object for the Web Agent protecting the IIS Web Server. This object verifies that the SetRemoteUser parameter is set to yes to preventing any anonymous user from writing to the file system.

Configure the FWS Properties File at the IdP

The affwebservices.properties file contains all the initialization parameters for Federation Web Services. Modify at least one of the settings in this file.

To modify the affwebservices.properties file

1. On the IdP system with the Web Agent Option Pack, go to the directory C:\Program Files\ca\webagent\affwebservices\WEB-INF\classes
2. Set the AgentConfigLocation parameter to the location of the WebAgent.conf file. This parameter must have a value.

   For this deployment, an IIS web server hosts the FWS application. So, the path to the WebAgent.conf file is:

   ```
   C:\\Program Files\\ca\\webagent\\bin\\IIS\\WebAgent.conf
   ```

   Note: Federation Web Services is a Java component, so the Windows paths must contain double backslashes. This format applies only to Windows.

   Verify that this path is entered on one line.
3. Save and close the file.
4. Test Federation Web Services at the IdP (see page 39).

Test Federation Web Services at the IdP

After you set up Federation Web Services, verify that the application is operating correctly.

Follow these steps:

1. Open a web browser and enter the following link:

   http://<*fqhn*>:<*port_number*>/affwebservices/assertionretriever

   fqhn

   Defines the fully qualified host name.

   port_number

   Defines the port number of the server where the Web Agent and Web Agent Option Pack are installed.

   For this deployment, enter:

   http://www.idp.demo:80/affwebservices/assertionretriever

   If Federation Web Services is operating correctly, the following message appears:

   ```
   Assertion Retrieval Service has been successfully initialized.

   The requested servlet accepts only HTTP POST requests.
   ```

   This message indicates that Federation Web Services is listening for data activity. If Federation Web Services is not operating correctly, you get a message that the Assertion Retrieval Service has failed. If Assertion Retrieval Service fails, examine the Federation Web Services log.

2. Enable Web Agent Option Pack Logging at the IdP (see page 39).

Enable Web Agent Option Pack Logging at the IdP

At the IdP, enable logging for the system with the Web Agent Option Pack. You want to be able to view the following logs:

- affwebservices.log
- FWSTrace.log

Follow these steps:

1. Configure the affwebservices.log by setting up the LoggerConfig.properties file.
2. Configure FWS trace logging.
3. Specify the User Store for the IdP Policy Server.

Specify the User Store for the IdP Policy Server

The IdP user directory consists of user records for which the Identity Provider generates assertions.

The following steps specify how to configure a user directory in the Administrative UI. The directory IdP LDAP, is the Sun ONE LDAP directory that contains user1 and user2.

Follow these steps:

1. Log in to the Administrative UI.
2. Click Infrastructure, Directory, User Directories.
3. Click Create User Directory.
4. Complete the following fields:

   Name

   IdP LDAP

   NameSpace

   LDAP

   Server

   www.idp.demo:42088

5. Complete the following field in the LDAP Settings section:

   Root

   dc=idp,dc=demo

   Accept the defaults for the other values.

   Complete the following field in the LDAP User DN Lookup:

   Start

   uid=

   End

   ,ou=People,dc=idp,dc=demo

6. Click View Contents to verify you can view the contents of the directory.
7. Click Submit.
8. Set up an Affiliate Domain at the IdP.

Set up an Affiliate Domain at the IdP

To identify the Service Provider to the Identity Provider, create an affiliate domain and add a service provider object for sp.demo.

Follow these steps:

1. Log in to the Administrative UI.
2. Click Federation, Legacy Federation, Affiliate Domains.
3. Click Create Affiliate Domain.
4. Complete the following fields:

   Name

   Federation Sample Partners

   Description

   Domain for sp.demo
5. Leave this dialog open and add the user directory to the affiliate domain at the IdP (see page 41).

Add the User Directory to the Affiliate Domain at the IdP

Associate a user directory with the affiliate domain.

Follow these steps:

1. Complete the User Directory section in the Affiliate Domain dialog.
2. Add the IdP LDAP directory.

   For your network, select the user store you set up at the IdP.
3. Click OK.
4. Go to Add the Service Provider to the Affiliate Domain at the IdP (see page 42).

Add the Service Provider to the Affiliate Domain at the IdP

Add the Service Provider named sp.demo to the affiliate domain.

Follow these steps:

1. In the Administrative UI, navigate to Federation, Legacy Federation, SAML Service Providers.
2. Select Create SAML Service Provider.
3. Follow the configuration wizard.
4. Select Federation Sample Partners as the domain then click Next.
5. Complete the following fields in the General step:

   Name

   sp.demo

   Description

   Service Provider

   SP ID

   sp.demo

   IdP ID

   idp.demo

   Skew Time (seconds)

   Accept the default

   Authentication URL

   http://www.idp.demo/siteminderagent/redirectjsp/redirect.jsp

   This redirect.jsp is included with the Web Agent Option Pack that is installed at the Identity Provider site. In this deployment, that server is www.idp.demo. If the user does not have a CA SiteMinder® session, the SSO service at the IdP redirects the user to the authentication URL to log in.

   After successful authentication, the redirect.jsp application redirects the user back to the SSO service for assertion generation. A CA SiteMinder® policy must protect this URL.

   Enabled

   Verify that this option is selected. By default, this option is selected.
6. Keep the UI open and go to Select Users for which the IdP Generates Assertions.

Protect the Authentication URL (SAML 2.0)

You must protect the Authentication URL with a SiteMinder policy. Protecting the Authentication URL ensures that a user requesting a protected federated resource is presented with an authentication challenge if they do not have a SiteMinder session at the IdP.

Follow these steps:

1. From Domains, create a policy domain called Authentication URL Protection Domain.
2. Add the IdP LDAP user directory in the User Directories page.
3. From the Authentication URL Protection domain, create a persistent realm with the following field entries:

   Name

   Authentication URL Protection Realm

   Agent

   Using the lookup button, select FSS web agent

   This is the Web Agent protecting the server with the Web Agent Option Pack.

   Resource Filter

   /siteminderagent/redirectjsp/redirect.jsp

   Accept the defaults for the other settings.

   Session tab

   Select Persistent Session
4. From the IDP Authentication URL Protection Realm, create a rule under the realm with the following field entries:

   Name

   Authentication URL Protection Rule

   Realm

   Authentication URL Protection Realm

   Resource

   *

   Web Agent actions

   Get

   Accept the defaults for the other settings.

5. From the Authentication URL Protection domain, create a policy with the following entries:

   Name

   Authentication URL Protection Policy

   Users tab

   Add user1 from the IdP LDAP user directory

   Rules tab

   add Authentication URL Protection Rule

   You now have a policy that protects the Authentication URL at the Identity Provider.

Select Users for which the IdP Generates Assertions

When you specify a Service Provider in an affiliate domain, include a list of users and groups for which the Assertion Generator generates SAML assertions. Add only users and groups from directories that are in an affiliate domain.

To select users for assertion generation

1. Navigate to the Users step.
2. In the User Directories section, select Add Members for the LDAP user directory previously configured.

   The Users/Groups dialog opens.
3. Search for user1 and user2 by completing the following fields:

   Search type

   Attribute-value

   Attribute

   uid

   Value

   *

   These employees are listed in the IdP LDAP.
4. Click OK.
5. Go to the next step in the wizard to configure a Name ID for the assertion.

Configure a Name ID for the Assertion

The Name ID is a unique way of identifying a user in an assertion. The NameID that you enter in the Administrative UI is included in the assertion.

To configure name IDs

1. Navigate to the Name IDs step.

   The Name IDs dialog displays.

2. Complete the following fields:

   Name ID Format

   Email Address

   The email address format value means that the Name ID must use an email address in the user directory to identify the user.

   Name ID Type section

   User Attribute

   Name ID Fields—Attribute Name

   mail

3. Keep the ui open go to the next step in the wizard.

Configure POST Single Sign-on at the IdP

Specify the HTTP-POST as the SAML 2.0 binding for single sign-on.

Follow these steps:

1. Navigate to the SAML Profiles step.
2. Complete the following fields:

   Audience

   sp.demo

   AuthnContext Class Ref

   urn:oasis:names:tc:SAML:2.0:ac:classes:Password (default)

   Assertion Consumer Service

   `http://www.sp.demo:81/affwebservices/public/saml2assertionconsumer`

   Specifies the URL of the Assertion Consumer Service. For your network, the server you specify is the SP web server where the Web Agent Option Pack is installed.

Authentication Level

5 (default)

Validity Duration Second(s)

60 (default)

In a test environment, if the following message appears in the Policy Server trace log, increase the Validity Duration value above 60.

```
Assertion rejected(_b6717b8c00a5c32838208078738c05ce6237) –current time
(Fri Sep 09 17:28:33 EDT 2005) is after SessionNotOnOrAfter time (Fri Sep 09
17:28:20 EDT 2005)
```

HTTP-POST

Select this check box

3. Disregard the remaining fields.
4. Go to the next step in the wizard.

Disable Signature Processing for the Basic Sample Deployment

In a production environment, signature processing to sign assertions is required. However, for the basic sample deployment, disable signature processing.

Important! Never disable signature processing in a SAML 2.0 production environment.

Follow these steps:

1. Navigate to the Encryption&Signing step.
2. In the Signature section of the page, select Disable Signature Processing.
3. Click Next to move to the Attributes step in the wizard.

Complete the Service Provider Object Configuration

Attributes is the final step in Service Provider configuration. For a basic configuration, do not configure attributes. Instead, click Finish to complete the Service Provider configuration. The configuration is submitted. You have identified a Service Provider object for the Identity Provider.

Configure the Service Provider

After completing the configuration at the Identity Provider, you must Set Up the Service Provider.

Set up the Service Provider for the Sample Network

To deploy legacy federation at the Service Provider, the following sections detail the tasks. The entries in each section reflect the sample data provided for a basic configuration.

Note: These procedures assume you have already installed the required components.

Set Up the SP User Store

At the SP, configure a user store and add user records for users that require assertions. When the assertion is presented during authentication, the Service Provider looks in the user store for the user record.

In this deployment, the Sun ONE LDAP user directory is the user store. Use the Sun ONE Server Console to add users to the directory.

To configure the user store

1. Add the following users:
   - user1
   - user2
2. Fill in the attributes for user1 and user2 as follows:

   user1

   userpassword: customer

   mail: user1@sp.demo

   user2

   userpassword: customer

   mail: user2@sp.demo

   Important! The email address must be the same in the Identity Provider user store for the same users.
3. Enable trace logging (see page 49).

Point the Policy Server to the SP LDAP Policy Store

Establish the connection between the Policy Server and the LDAP policy store.

Follow these steps:

1. Open the Policy Server Management Console.
2. Select the Data tab.

   Complete the following fields:

   Databases

   Policy Store

   Storage

   LDAP

   LDAP IP Address

   sp.demo:389

   Root DN

   o=sp.demo

   Admin Username

   cn=Directory Manager

   Password

   federation

   Confirm Password

   federation
3. Click OK.
4. Set up the SP user store (see page 47).

Enable Trace Logging for Federation Components at the SP

At the SP Policy Server, configure the SiteMinder Profiler to log federation components to the trace log, smtracedefault.log and examine trace messages.

To enable logging

1. Open the Policy Server Management Console.
2. Click on the Profiler tab and customize the contents of the trace log. Be sure to include the Fed_Server component in the log to see the federation trace messages.

   To configure trace logging at the Policy Server, using the Policy Server Management Console.
3. Install the SP Web Agent.

Configure the Web Server with the Web Agent Option Pack

The Web Agent Option Pack installed the Federation Web Services (FWS) application. Configure the FWS application for the sample deployment.

For FWS to work, do the following

1. Install the JDK for Federation Web Services (see page 49)
2. Install and Configure ServletExec to Work with FWS at the SP (see page 49)
3. Configure the AffWebServices.properties file (see page 51)
4. Enable Web Agent Option Pack logging (see page 52)
5. Test Federation Web Services (see page 51)

Install the JDK for Federation Web Services

The Web Agent Option Pack requires a JDK to run the Federation Web Services application. For the specific version required, go the Technical Support site and search for SiteMinder Platform Support Matrix for the release.

Install and Configure ServletExec to Work with FWS at the SP

For FWS to operate in this deployment, ServletExec is installed on a Sun ONE 6.1 web server.

Note: CA SiteMinder® 12.52 SP1 is shipped with a ServletExec license key file named ServletExec_AS_6_license_key.txt. If you do not have this license key, contact CA Technical Support. From this license file, copy the license key and enter it in the ServletExec License dialog of the ServletExec Administration Console. For instructions on licensing ServletExec, see ServletExec documentation, available at the New Atlanta Communication http://www.newatlanta.com website.

Apply the most current hot fixes for the supported version of ServletExec. The hot fixes are necessary for Federation Web Services to work with ServletExec. To obtain the hot fixes, go to the website for New Atlanta Communications http://www.newatlanta.com.

To set up ServletExec

1. Install ServletExec.

   For instructions, refer to New Atlanta Communications documentation.

2. Open the ServletExec Administration Console.

3. Under Web Applications, select manage.

   The Manage Web Applications dialog opens.

4. Click Add a Web Application.

5. Enter the following information:

   Application Name

   affwebservices

   URL Context Path

   /affwebservices/

   Location

   C:\program files\ca\webagent\affwebservices

   The location of affwebservices in your network can be different. Enter the correct location.

6. Click Submit.

7. Exit the ServletExec Console.

8. Configure the AffWebServices.properties file (see page 51).

Configure the FWS Properties File

The AffWebServices.properties file contains all the initialization parameters for Federation Web Services. Specify the location of the WebAgent.conf file in this file.

Follow these steps:

1. On the SP system with the Web Agent Option Pack, go to the directory C:\Program Files\ca\webagent\affwebservices\WEB-INF\classes
2. Set the AgentConfigLocation parameter to the location of the WebAgent.conf file. Setting a value for this parameter is mandatory.

   For this deployment, the web server hosting the FWS application at the Service Provider is a Sun ONE Web Server. So, the path to the WebAgent.conf file is:

   ```
   C:\\Sun\\WebServer6.1\\https-sp.demo\\config\\WebAgent.conf
   ```

   Note: Federation Web Services is a Java component, so the Windows paths must contain double backslashes. Specify this entry on one line.
3. Save and close the file.
4. Test Federation Web Services (see page 51).

Test Federation Web Services

After you have set up the Federation Web Services application, verify that it is operating properly.

Follow these steps:

1. Open a web browser and enter the following link:

   `http://`*`fqhn`*`:`*`port_number`*`/affwebservices/assertionretriever`

   fqhn

   Defines the fully qualified host name.

   port_number

   Defines the port number of the server where the Web Agent and Web Agent Option Pack are installed.

   For this deployment, enter:

   ```
   http://www.sp.demo:81/affwebservices/assertionretriever
   ```

If Federation Web Services is operating correctly, the following message appears:

```
Assertion Retrieval Service has been successfully initialized.
The requested servlet accepts only HTTP POST requests.
```

This message indicates that Federation Web Services is listening for data activity. If Federation Web Services is not operating correctly, you get a message that the Assertion Retrieval Service has failed. If Assertion Retrieval Service fails, examine the Federation Web Services log.

2. Enable Web Agent Option Pack logging. (see page 52)

Enable Web Agent Option Pack Logging at the SP

At the SP, enable logging for the system with the Web Agent Option Pack so you can view the following logs:

- affwebserv.log

  Contains error logging messages.

- FWSTrace.log

To enable error and trace logging

1. Open up the LoggerConfig.properties file. This file can be found in the directory *web_agent_home*/affwebservices/WEB-INF/classes.
2. Set the LoggingOn parameter to Y.
3. Accept the default name and location for the LogFileName setting, which points to the affwebserv.log file.
4. Set the TracingOn setting to Y.
5. Accept the default name and location for the TraceFileName setting, which points to the FWSTrace.log file.

Logging is now enabled.

Specify the User Store for the SP Policy Server

The SP user directory consists of user records for which the Service Provider uses for authentication.

Configure a user directory in the Administrative UI. The directory, named SP LDAP, is the Sun ONE LDAP directory that contains the users user1 and user2.

Follow these steps:

1. Log in to the Administrative UI.
2. Click Infrastructure, Directory, User Directories.
3. Click Create User Directory.
4. Complete the following field:

   Name

   SP LDAP
5. Complete the following fields in the Directory Setup section:

   Namespace

   LDAP

   Server

   www.sp.demo:32941
6. Complete the following fields in the LDAP Search section:

   Root

   dc=sp,dc=demo

   Accept the defaults for the other values.
7. Complete the following fields in the LDAP User DN Lookup section:

   Start

   uid=

   End

   ,ou=People,dc=sp,dc=demo
8. Click View Contents to verify that you can view the contents of the directory.
9. Click Submit.

Configure the SAML 2.0 Authentication Scheme at the SP

To authenticate users at the Service Provider, configure the SAML 2.0 authentication scheme. The assertion from the IdP provides the credentials for authentication.

Follow these steps:

1. Log in to the Administrative UI.
2. Click Infrastructure, Authentication, Authentication Schemes.

3. Complete the following fields:

   Scheme Common Setup section:

   Name

   Partner IDP.demo Auth Scheme

   Authentication Scheme Type

   SAML 2.0 Template

   Protection Level

   5 (default)

4. Click SAML 2.0 Configuration.

   The dialog where you specify the general and user disambiguation displays.

5. Specify the following settings in the General section:

   SP ID

   sp.demo

   IdP ID

   idp.demo

   SAML Version

   2.0 (default)

   Skew Time

   30 (default)

   Note: The SP ID and IdP ID values must match the values at the IdP.

6. In the User Disambiguation section, configure the following setting:

   LDAP

   Username=%s

7. Click Next to move to the single sign-on settings.

More information:

Enable Signature Validation at the Service Provider (see page 70)

Configure HTTP-POST for Single Sign-on at the SP

For the authentication scheme, indicate the single sign-on binding to be used so the Service Provider knows how to communicate with the Identity Provider.

Follow these steps:

1. In the SSO settings, complete the following fields:

   Redirect Mode

   302 Cookie Data (default)

   User is redirected through an HTTP 302 redirect with a session cookie, but no other data.

   SSO Service

   `http://www.idp.demo:80/affwebservices/public/saml2sso`

   Audience

   sp.demo

   This value must match the value at the Identity Provider.

   Target

   `http://www.sp.demo:81/spsample/protected/target.jsp`

   If you begin the Target with http, enter the full path to the resource. A CA SiteMinder® policy that uses the SAML 2.0 authentication scheme protects the target.

2. Select the HTTP-POST in the Bindings section.

3. Clear the check box labeled Enforce Single Use Policy.

   Disabling this option makes the sample network noncompliant with SAML 2.0. To enable the use of the single use policy feature, set up a session store at the Service Provider.

4. Click Next until you reach the Encryption & Signing step.
5. Select Disable Signature Processing.

   Important! Disabling signing is intended *only* for debugging the initial single sign-on configuration. In a production environment, signature processing is a mandatory security requirement. At the SP, enable signature validation and set up the certificate data store to validate signatures.
6. Click Next until you reach the last configuration step.
7. Click Finish.

   The basic authentication scheme configuration is complete.
8. Keep the Administrative UI open and go to Protect the Target Resource Using SAML 2.0 Authentication.

Protect the Target Resource at the SP

After you configure a SAML 2.0 authentication scheme, use this scheme in a policy that protects the target resource at Service Provider.

Follow these steps:

1. Navigate to Infrastructure, Agent, Agents and create a Web Agent named sp-webagent. This Agent protects the server with the Web Agent Option Pack installed.
2. Navigate to Policies, Domain, Domains.
3. Create a policy domain with the following values:

   Name

   Domain for IdP.demo Visitors

   User Directory section

   Add the user directory that holds user1 and user2.
4. Go to the Realms page and configure a persistent realm with the following values:

   Name

   SP Target Page Protection Realm

   Agent

   sp-webagent

   Resource Filter

   /spsample/protected.jsp

   Defines the path to the target resource at the Service Provider web server.

Default Resource Protection

Protected

Authentication Scheme

Partner IdP.demo Auth Scheme

5. To the realm, add a rule with the following values:

Name

SP Target Page Protection Rule

Realm

SP Target Page Protection Realm

Resource

*

Action

Web Agent actions

Get

Accept the defaults for all other fields.

6. Go to the Policies page and create a policy with the following values:

General page

Name

SP Target Page Protection Policy

Users pagexs

For the SP LDAP directory, click Add Member. Add user1 so this user has access to the target.

Rules page

Add the SP Target Page Protection Rule

7. Click Submit.

The protection policy for the target resource is complete.

8. Exit the Administrative UI.
9. Use HTML Pages to test the federation set-up (see page 58).

Test SAML 2.0 Single Sign-on

To test single sign-on in a CA SiteMinder®-to-CA SiteMinder® network, use your own HTML page. The HTML page must contain a hard-coded link to the AuthnRequest service. For this deployment, the sample link for POST binding is:

```
http://www.sp.demo:81/affwebservices/public/saml2authnrequest?ProviderID=idp.demo
```

The AuthnRequest Service redirects the user to the Identity Provider specified in the link to retrieve the authentication context of the user. After the Identity Provider authenticates the user and establishes a session, it directs the user back to the target resource at the Service Provider.

Note: The ProviderID in the Authnrequest link must match the IdP ID field value specified in the SAML authentication scheme at the SP. The IdP ID field is on the Scheme Setup tab of the Authentication Scheme Properties dialog.

To test federated single sign-on

1. Open up a browser.
2. Enter the URL for the web page that has links to trigger single sign-on.

   A login challenge has to display.
3. Using the login of an existing user in your user store, enter the user credentials. For example, if user1 is a user in the user store, enter the credentials for this user.

   If single sign-on is successful, your target page displays.
4. After you test single sign-on, you can Add Functionality to the Federation Deployment (see page 58).

Add Functionality to the Federation Deployment

After you complete the POST single sign-on configuration, you can add more features to the federated network.

The additional tasks covered in this deployment example are:

- Configuring single logout
- Configuring artifact single sign-on
- Adding an attribute to an assertion
- Enabling digital signing of an assertion
- Encrypting and decrypting an assertion

Some of these additional features are required for single sign-on in a production environment, such as digital signing for POST binding. Required tasks are noted.

Configure Single Logout

The single logout protocol (SLO) results in the simultaneous end of all sessions for a particular user, to help ensure security. Associate these sessions with the browser that initiated the logout. Single logout does not necessarily end all sessions for a user.

Configuring single logout enables the Identity Provider and Service Provider to support the single logout protocol. The configuration also determines how single logout is handled.

Enable Single Logout at the IdP

You can initiate single logout at the IdP. At the IdP, idp.demo, enable single logout on a per-SP basis.

Follow these steps:

1. Log in to the Administrative UI and access the SAML Service Provider object for sp.demo.
2. Navigate to the SAML Profiles page.
3. Select HTTP-Redirect.

   The remaining fields become active.
4. Enter values for the following fields:

   SLO Location URL

   http://www.sp.demo:81/affwebservices/public/saml2slo

   Defines the SLO servlet at the SP.

   SLO Confirm URL

   http://www.idp.demo:80/idpsample/SLOConfirm.jsp.
5. Accept defaults for the other fields.
6. Click Submit.
7. Log in to the Policy Server Management Console and enable the session store.

   For instructions, see the *Policy Server Administration Guide*.

Enable Single Logout at the SP

You can initiate single logout at the Service Provider.

Follow these steps:

1. Verify that the realm with the protected resources is configured for persistent sessions.
2. Navigate to the authentication scheme named Partner IDP.demo Auth Scheme.
3. Modify the SAML 2.0 Configuration for the scheme to access the SLO tab.
4. In the SLO tab, select HTTP-Redirect.

   The rest of the fields become active.
5. Complete the fields as follows:

   SLO Location URL

   http://www.idp.demo:80/affwebservices/public/saml2slo

   SLO Confirm URL

   http://www.sp.demo:81/spsample/SLOConfirm.jsp
6. Accept the default values for all other fields.
7. Log in to the Policy Server Management Console and enable the session store.

   For instructions, see the *Policy Server Administration Guide*.

Test Single Logout

Use your own web pages to test single logout. Verify that your HTML page for testing SP-initiated single sign-on includes a hard-coded link to the single logout service.

After you successfully test single sign-on, you can test single logout. From the landing page that you created, click a link that directs the browser to the single logout URL using the HTTP redirect binding.

Configure SAML 2.0 Artifact Single Sign-on

Complete tasks at the Identity Provider and Services Provider to configure artifact single sign-on.

Required tasks at the Identity Provider:

- Set up the IdP session store (see page 61)
- Enable SSL for the IdP web server (see page 62)
- Permit access to the artifact resolution service policy
- Enable a persistent session to store assertions at the IdP (see page 63)
- Select the artifact binding at the IdP

Required tasks at the Service Provider:

- Add a CA certificate to the certificate data store at the SP
- Enable the artifact binding at the SP
- Test artifact single sign-on (see page 67)

Set Up the IdP Session Store for Artifact Single Sign-on

For artifact binding, set up and enable the session store at the IdP. When you use the artifact binding, the session store is required to store the assertion before it is retrieved with the artifact.

To enable the session store

1. Install and configure an ODBC database to serve as the session store. In this deployment, we are using Microsoft SQL Server.

   For instructions, see the *Policy Server Installation Guide*.
2. Open the Policy Server Management Console.
3. Select the Data tab.
4. Select Session Server From the Database drop-down list.
5. Complete the following fields:

   Data Source Information

   SiteMinder Session Data Source

   User Name

   admin

Password

dbpassword

Confirm Password

dbpassword

Maximum connections

16 (default)

6. Select the Enable Session Server check box.
7. Click OK to save the settings.
8. Enable SSL for the IdP Web Server for Artifact Single Sign-on (see page 62).

Enable SSL for the IdP Web Server for Artifact Single Sign-on

Enable SSL for the web server where the Web Agent Option Pack is installed. Enabling SSL verifies that the back channel over which the assertion is passed is secure.

Follow these steps:

1. Create a server-side certificate request.
2. Have the Certificate Authority sign the server-side certificate.
3. Specify the server-side certificate in the web server configuration.

   For the IIS Web Server used in the sample network, the IIS Certificate Wizard would be used.
4. Enable a Persistent Session to Store Assertions at the IdP.

Permit Access to FWS Policy for the Artifact Resolution Service

The Web Agent Option Pack installs the Federation Web Services application (FWS). When you install the Policy Server for the same IdP as the Web Agent, several policies for services within the FWS application are automatically created. One of these policies protects the artifact resolution service for HTTP-Artifact single sign-on.

Specify which relying partners can access the artifact resolution service by enforcing protection of this artifact resolution policy.

Follow these steps: at the IdP

1. Log on to the Administrative UI.
2. Click Infrastructure, Agent, Agents.
3. Click Create Agent.
4. Enter idp-webagent in the Name field, which is the name of the agent in this sample deployment. Click Submit.

5. Select Infrastructure, Agent Groups.
6. Select the FederationWebServicesAgentGroup entry.

   The Agent Groups dialog opens.
7. Click Add/Remove and the Agent Group Members dialog opens.
8. Move idp-webagent from the Available Members list to the Selected Members list.
9. Click OK to return to the Agent Groups dialog.
10. Click Submit then click Close to return to the main page.
11. Specify that all Service Providers in the affiliate domain Federation Sample Partners can access the artifact resolution service, as follows:
    a. Select Infrastructure, Policies, Domain, Domains.
    b. Select the FederationWebServicesDomain.
    c. Select the Policies tab, then click Modify.
    d. From the Policy List, click the Edit arrow to the right of the SAML2FWSArtifactResolutionServicePolicy entry.

       The Policy dialog opens.
    e. From the Users tab, select Add Members for the SAML2FederationCustomUserStore directory.

       The Users/Groups dialog opens.
    f. Select affiliate:FederationSamplePartners from the list and click OK.
    g. Click Submit to complete the changes.

The policy that protects the artifact resolution service is now being enforced.

Enable a Persistent Session to Store Assertions at the IdP

Enable a persistent session for the realm that contains the protected authentication URL (Protect the Authentication URL). The persistent session is required to store assertions for SAML artifact binding.

If you did not enable a persistent session when you protected the authentication URL, enable it now.

Follow these steps:

1. Log in to the Administrative UI.
2. Navigate to Infrastructure, Domain, Domains.
3. Access the domain for the authentication URL. realm.

4. From the Realms page, select the realm with the authentication URL and modify it.
5. In the Session section, select Persistent.
6. Click OK.
7. Select the Artifact Binding at the IdP.

Select the Artifact Binding at the IdP

For artifact single sign-on, enable the artifact binding.

Follow these steps:

1. Click Federation, Legacy Federation, SAML Service Providers.
2. Select sp.demo to access the settings for this partner.
3. Click Modify and select the SAML Profiles page.
4. Complete the following fields:

   Audience

   sp.demo

   This value must match the value at the Service Provider.

   Assertion Consumer Service

   ```
   http://www.sp.demo:81/affwebservices/public/
   saml2assertionconsumer
   ```

   Authentication Level, Validity Duration, AuthnContext Class Ref

   accept the defaults

   In a test environment, you can increase the Validity Duration value above 60, the default, if you see the following message in the Policy Server trace log:

   ```
   Assertion rejected (_b6717b8c00a5c32838208078738c05ce6237) – current time
   (Fri Sep 09 17:28:33 EDT 2005) is after SessionNotOnOrAfter time (Fri Sep 09
   17:28:20 EDT 2005)
   ```

5. In the Artifact Bindings section, select HTTP-Artifact

6. For the Artifact Encoding field, select URL.

   The artifact is added to a URL-encoded query string.

7. Go to the Attributes page.

8. In the Backchannel section, complete the following fields:

   Password

   smfederation

   Confirm Password

   smfederation

   The Identity Provider uses this password for secure communication across the backchannel.

9. Click Submit.

10. Add a CA certificate to the certificate data store at the SP.

Add a CA Certificate for an SSL Back Channel at the SP

For artifact single sign-on, if Basic over SSL is the authentication scheme protecting the Artifact Resolution Service, add a certificate to the certificate data store at the Service Provider.

The certificate data store holds the Certificate Authority certificate that establishes an SSL connection between the Service Provider and the Identity Provider. The certificate secures the back channel that the assertion is sent across. Protect the Artifact Resolution Service and secure the back channel so the Service Provider knows that a trusted authority secures the SSL connection.

A set of common root and intermediate CAs are included with CA SiteMinder®. To use CA certificates that are not in the certificate data store, import them.

For this deployment, the alias is sampleAppCertCA and the certificate of the CA is docCA.crt.

Follow these steps:

1. Log in to the Administrative UI.

2. Click Infrastructure, X509 Certificate Management, Certificate Authorities.

3. Click Import New.

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

4. At the Select File step, select docCA.crt.

   The wizard skips the Password step.

5. In the Select Entries step, enter sampleAppCertCA in the Alias column.

6. At the Confirm step, review the certificate information and click Finish.

   The CA certificate is imported into the certificate data store. The change takes place directly after the import is complete.

7. Enable the Artifact Binding for SAML Authentication at the SP.

Important! You cannot delete a CA certificate that is part of a trust chain for other certificates the system uses. If you try to delete a CA certificate in use, an error message states that the certificate cannot be deleted.

Enable the Artifact Binding at the Service Provider

At the Service Provider, configure the single sign-on bindings for the SAML authentication scheme. The configuration instructs the Service Provider how to communicate with the Identity Provider.

Follow these steps:

1. Click Infrastructure, Authentication, Authentication Schemes.
2. Select the Partner IDP.demo Auth Scheme. This authentication scheme is the one you created for the basic configuration.
3. Select Modify, SAML 2.0 Configuration, SSO tab.
4. Enter the following value for the Resolution Service field:

   https:/www.idp.demo:443/affwebservices/saml2artifactresolution

5. Go to the Encryption & Signing page.
6. In the Backchannel section, complete the following fields:

   Authentication

   Basic

   SP Name

   sp.demo

   Password

   smfederation

   Confirm Password

   smfederation

   The password must match the password at the Identity Provider. This password enables secure access across the back channel to the artifact resolution service at the Identity Provider.

7. Click OK.
8. Test SP-initiated artifact single sign-on (see page 67).

Test Artifact Single Sign-on

Test single sign-on in a CA SiteMinder®-to-CA SiteMinder® network using your own web pages.

Your own HTML page must contain a hard-coded link to the AuthnRequest service. For this deployment, the link for Artifact binding is:

```
http://<server:port>/affwebservices/public/saml2authnrequest?ProviderID=
IdP_ID&ProtocolBinding=urn:oasis:names:tc:SAML:2.0:bindings:HTTP-Artifact
```

server:port

Defines the name and port of the server at the SP where the Web Agent Option Pack is installed.

IdP_ID

Defines the provider ID.

The link for this deployment is:

```
http://www.sp.demo:81/affwebservices/public/saml2authnrequest?ProviderID=
idp.demo&ProtocolBinding=urn:oasis:names:tc:SAML:2.0:bindings:HTTP-Artifact
```

The HTML source file with the link is similar to the following example:

```
<a
href="http://www.sp.demo:81/affwebservices/public/saml2authnrequest?ProviderI
D=
idp.demo&ProtocolBinding=urn:oasis:names:tc:SAML:2.0:bindings:HTTP-Artifact">
Link for ARTIFACT Single Sign-on</a>
```

The AuthnRequest Service redirects the user to the Identity Provider specified in the link to retrieve the authentication context of the user. After the Identity Provider authenticates the user and establishes a session, it directs the user back to the target resource at the Service Provider.

Note: The ProviderID in the Authnrequest link must match the IdP ID field value at the SAML authentication scheme at the SP. The IdP ID field is on the Scheme Setup tab of the Authentication Scheme Properties dialog.

Now, follow the steps to test SP-initiated single sign-on (see page 58).

Include an Attribute in the Assertion

You can add attributes from the user store record to a SAML assertion to identify a user. The attribute must exist in the user store of the Identity Provider for that specific user who is requesting access to the target resource.

For this deployment, add an attribute for user1 that represents the givenname in the user record.

Follow these steps: at the Identity Provider

1. Log in to the Administrative UI.
2. Click Federation, Legacy Federation, SAML Service Providers.
3. Select sm.demo.
4. Click Modify then navigate to the Attributes page.
5. Click Add in the Attributes section.
6. In the Add Attributes dialog, complete the following fields:

   Attribute Type

   unspecified (default)

   Attribute Kind

   User Attribute

   Variable Name

   firstname

   Attribute Name

   givenname

   givenname is an attribute in the profile of user1.
7. Click OK to save your changes and return to the Attributes page.
8. Click Submit.

Configure Digital Signing and Verification

For SAML 2.0 POST single sign-on, the Identity Provider must sign the SAML response. The configuration tasks at the Identity Provider enable digital signing. The configuration tasks at the Service Provider enable signature verification.

Important! In a production environment, signature processing is a mandatory security requirement.

- Enable Signing at the Identity Provider (see page 69)
- Enable Signature Verification at the Service Provider (see page 70)

Enable Signing at the Identity Provider

Keys and certificates that sign SAML assertions for POST binding are stored in the certificate data store. Signing a SAML response is required, so the certificate data store at the Identity Provider must contain the appropriate key/certificate pair.

Deploy the sample application using the key/certificate pair that it automatically installs. If you want to import a new key/certificate pair, complete the following procedures.

To import a private key/certificate pair

1. Log in to the Administrative UI.
2. Navigate to Infrastructure, X509 Certificate Management, Trusted Certificates and Private Keys.
3. Import the private key/certificate idp.demo to the certificate data store.

   idp.demo signs the SAML response.
4. Select Federation, Legacy Federation, SAML Service Providers.
5. Select the Service Provider, sp.demo then go to the Encryption & Signing tab.
6. Clear the Disable Signature Processing check box. Deselecting this check box means that signature processing is enabled.
7. Complete the following fields:

   Signing Alias

   Enter the alias that you specified when importing the private key/certificate pair.

   Signing Algorithm

   RSAwithSHA1 (default)

   POST Signature Options

   Sign Assertion (default)
8. Click Submit.

Enable Signature Validation at the Service Provider

For POST single sign-on, the Identity Provider must digitally sign the SAML assertion. Consequently, the Service Provider must validate the signature.

To validate a digital signature

- Import the certificate (public key) to the certificate data store.
- Specify the DN of the issuer and serial number of the certificate.

To import the public key

1. In the Administrative UI, navigate to Infrastructure, X509 Certificate Management, Trusted Certificates and Private Keys.
2. Add the public key/certificate pair to the certificate data store. In this deployment, the certificate is post-cert.crt.

   The key/certificate pair is added to the data store.
3. Click Submit.
4. Navigate to Infrastructure, Authentication, Authentication Schemes.
5. Select the SAML 2.0 authentication scheme, Partner IdP.demo Auth Scheme.
6. Select the Encryption & Signing tab.
7. In the D-Sign Info section, clear the Disable Signature Processing check box to enable signature processing.
8. Specify the following field values:

   Issuer DN

   CN=Certificate Manager,OU=IAM,O=CA.COM

   Serial Number

   008D 8B6A D18C 46D8 5B

   The D-Sig information enables the Service Provider to verify the SAML response signature. The values for the Issuer DN and Serial Number are from the certificate stored in the certificate data store at the Service Provider.
9. Click OK.

   Validation configuration is now complete.
10. Test POST single sign-on.

Encrypt and Decrypt the Assertion

For added security, you can encrypt the assertion. Encryption is an optional task that can be performed after you have configured a basic single sign-on network.

The Identity Provider encrypts the assertion with the certificate that corresponds to the private key/certificate pair that the Service Provider uses to decrypt the assertion.

The configuration tasks are available at the Identity Provider and Service Provider.

- Enable Encryption at the Identity Provider (see page 71)
- Enable Decryption at the Service Provider (see page 72)

Enable Encryption of the Asserton at the IdP

In this deployment, sp_encrypt.crt is the certificate for encryption.

To enable encryption at the IdP

1. Log in to the Administrative UI.
2. Navigate to Infrastructure, X509 Certificate Management, Trusted Certificates and Private Keys.
3. Import the sp-encrypt.crt certificate into the certificate data store.
4. Navigate to Federation, Legacy Federation, SAML Service Providers.
5. Select sp.demo.
6. Select Modify, SAML 2.0 Configuration.
7. Go to the the Encryption & Signing tab.
8. Select the Encrypt Assertion.
9. Accept the defaults for the Encryption Block Algorithm and the Encryption Key Algorithm.
10. In the Issuer DN, enter the issuer of the certificate. In this deployment, the DN is:

    CN=Doc Certificate Authority, OU=Doc, O=CA.COM

    Note: The value you enter for the Issuer DN field should match the issuer DN of the certificate in the certificate data store. View the DN to verify at you enter a matching value.
11. In the Serial Number field, enter the serial number of the certificate in the certificate data store. In this deployment, the value is 00EFF6AFB49925C3F4

    The number must be hexadecimal.
12. Click OK to save your changes.
13. Decrypt an Encrypted Assertion at the SP.

Enable Decryption of the Assertion at the SP

If the assertion is encrypted at the Identity Provider, the Service Provider must have the private key and corresponding certificate in the certificate data store.

The Service Provider accepts an encrypted assertion from the Identity Provider as long as it has the private key/certificate pair to decrypt the assertion.

Note: You do not have to enable the Require an Encrypted Assertion feature to accept an encrypted assertion at the Service Provider.

Follow these steps:

1. Open a command window.
2. Navigate to Infrastructure, X509 Certificate Management, Trusted Certificates and Private Keys.
3. Add the private key/certificate pair sp-encrypt.crt into the certificate data store. The alias for this pair is sp1privkey. The password for the private key is fedsvcs.
4. Test single sign-on. Go to either of the following:
   - Test SAML 2.0 POST Single Sign-on
   - Test Artifact Single Sign-on

Chapter 3: Overview of a CA SiteMinder® Federation Setup

This section contains the following topics:

Federation Setup Overview

This overview outlines the setup of a federated network.

The steps in each procedure are divided by tasks by the party generating assertions and tasks by the party consuming assertions. Within this organization, the procedures are further divided by the Policy Server and Web Agent tasks at each site.

For general purposes, this guide uses the terms asserting party and relying party to identify sides of a federated relationship.

The party that generates assertions is referred to as the *asserting party*. The asserting party can be any of the following partners:

- SAML 1.x producer
- SAML 2.0 Identity Provider
- WS-Federation Account Partner

The party that consumes assertions for authentication purposes is referred to as the *relying party*. The relying party can be any of the following partners:

- SAML 1.x consumer
- SAML 2.0 Service Provider
- WS-Federation Resource Partner

Note: You can perform all the installation tasks first then complete the software configuration through the Administrative UI.

These procedures refer to the latest CA SiteMinder® release. See the Platform Matrix for version information for a release.

To locate the Platform Support Matrix

1. Log on to the Technical Support Site.
2. Search for Platform Support Matrix.

Be aware of the following information:

- The product does not support federation between two systems using the same cookie domain.
- Legacy Federation is a separately licensed items from core CA SiteMinder®.

Conventions in the Installation Overview Procedures

The following variables are used in installation and configuration procedures:

web_agent_home

Specifies the installed location of the Web Agent

policy_server_home

Specifies the installed location of the Policy Server

web_server_home

Indicates the installed location of the web server

fqhn

Designates fully qualified host name

port_number

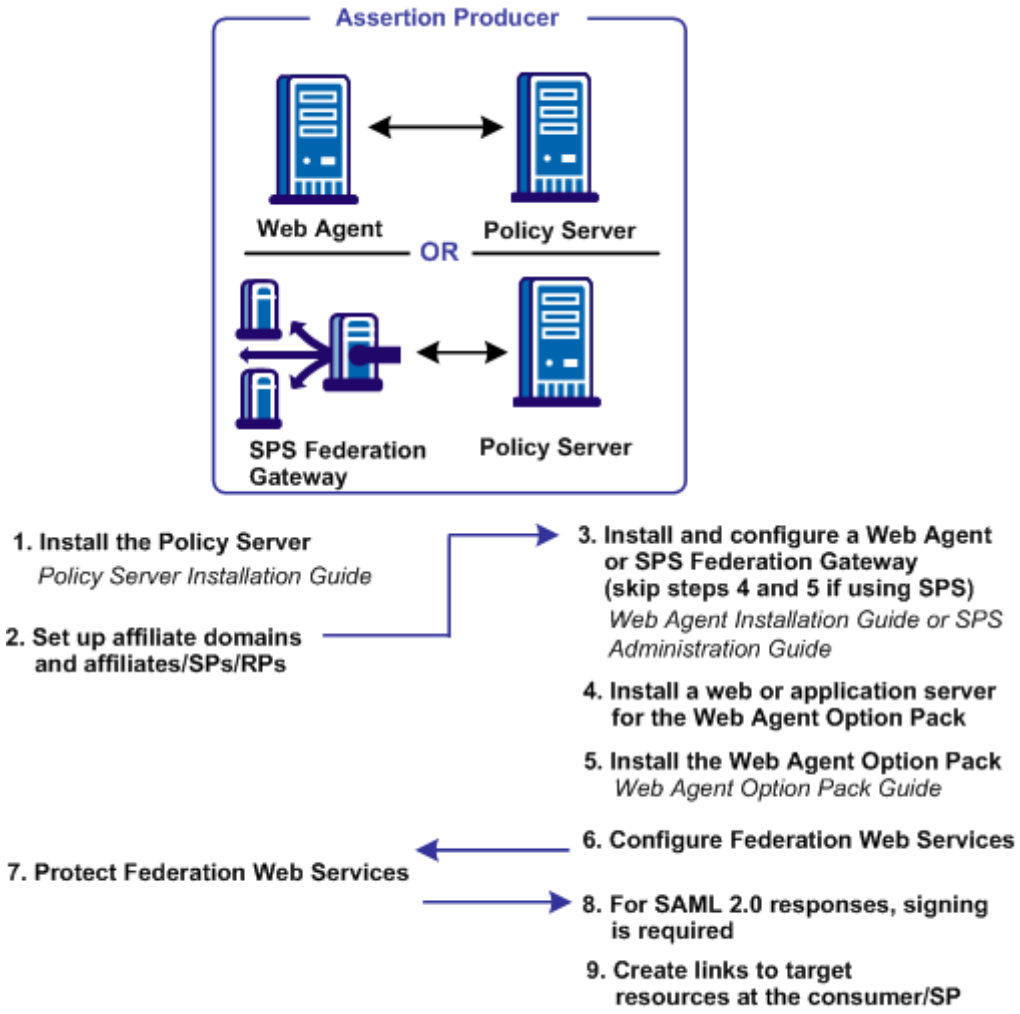
Specifies the port number of a server

sps_home

Specifies the installed location of CA SiteMinder® SPS

Set Up Asserting Party Components

The following illustration shows a SAML 1.x Producer, SAML 2.0 Identity Provider, or WS-Federation Account Partner setup.



Note: The SPS federation gateway can replace the Web Agent and Web Agent Option Pack to provide the Federation Web Services application functions. For information about installing and configuring the SPS federation gateway, see the *Secure Proxy Server Administration Guide*.

Install the Asserting Party Policy Server

The setup at the asserting party is as follows:

1. Install the Policy Server.

   See the *Policy Server Installation Guide*.

2. Set up the session store and its database for artifact single sign-on only.

   For information about the session store, see the *Policy Server Administration Guide.*

   The session store is required only for artifact single sign-on because the session server stores an assertion before it is retrieved.

3. Set up a policy store for use by the Policy Server. For instructions, see the *Policy Server Installation Guide*.

4. Set up a user directory, as instructed in the *Policy Server Configuration Guide*.

   This user directory must contain the users for which assertions are generated.

5. (Optional) Enable error and trace logging for the Policy Server to see the communication between the asserting and relying parties.

Set up Affiliate Domains and Add Sites to these Domains

Before you set up Federation Web Services, you establish affiliate domains and add the sites that consume assertions to the affiliate domains. The affiliate domains identify the partners to the site generating the assertions.

At the asserting party

1. Access the Administrative UI.
2. Create an affiliate domain.
3. Add a user store for users that the asserting party (producer, IdP, AP) generates assertions.
4. Add an object for each relying party (consumer, SP, RP) to the affiliate domain.

   There must be a one-to-one correspondence between a relying party and each object added to the domain.

5. After you add sites to an affiliate domain, verify that the Authentication URL is protected. This verification affirms that a user has a session at the asserting party before processing a request for a federated resource.

   To do this task:

   a. Create a policy domain.

   b. Protect the policy domain with the Web Agent. Use the Web Agent that is protecting the server with the Web Agent Option Pack.

   c. To this policy domain, add a realm, rule, and policy that protects the Authentication URL.

More Information:

Authenticate Users with No CA SiteMinder® Session (SAML 1.x) (see page 121)

Install a Web Agent or SPS Federation Gateway at the Asserting Party

The Web Agent is a required component in a CA SiteMinder® federation network. Install a Web Agent on a web server or install an SPS federation gateway, which has an embedded web agent.

At the asserting party, set up the following components:

1. Install one of the following components:

   - Web Agent

     For instructions, see the *Web Agent Installation Guide*.

   - SPS federation gateway

     For instructions, see the *Secure Proxy Server Administration Guide*.

2. For artifact single sign-on, SSL-enable the web server with the Web Agent installed or the system with the SPS federation gateway.

Install an Application Server for the Web Agent Option Pack (Asserting Party)

If you are implementing legacy federation with a Web Agent and Web Agent Option Pack, install the Web Agent Option Pack. Install this component on a web or application server.

At the asserting party:

1. Install one of the following servers to run Federation Web Services, the application that is installed with the Web Agent Option Pack.
   - Web server running ServletExec
   - WebLogic Application Server
   - WebSphere Application Server
   - JBOSS Application Server
   - Tomcat Application Server
2. Deploy Federation Web Services on these systems.
3. For artifact single sign-on, SSL-enable the web server where the Web Agent Option Pack is installed.

Install the Asserting Party Web Agent Option Pack

The Web Agent Option Pack supplies the Federation Web Services application, which is a required component for CA SiteMinder® legacy federation.

At the asserting party:

1. Install the Web Agent Option Pack.

   For instructions, see the *Web Agent Option Pack Guide*.
2. Verify that you installed a JDK. The Web Agent Option Pack requires a JDK.

   For the supported JDK version, log on to the Technical Support site and search for the CA SiteMinder® Platform Support Matrix for the release.

Note: The SPS federation gateway can replace the Web Agent and Web Agent Option Pack to provide the Federation Web Services application functions. For information about installing and configuring the SPS federation gateway, see the *Secure Proxy Server Administration Guide*.

Configure Federation Web Services (Asserting Party)

The Federation Web Services application is installed on the server with the Web Agent Option Pack or the SPS federation gateway.

To configure Federation Web Services at the asserting party

1. Configure one of the supported application servers to use the Web Agent Option Pack. Refer to the Web Agent Option Pack deployment instructions.

   On the SPS federation gateway, Federation Web Services is already deployed.

2. Verify that the AgentConfigLocation parameter in the AffWebServices.properties file is set to the full path to the WebAgent.conf file. Be sure that the syntax is correct and the path appears on one line in the file.

   The AffWebServices.properties file contains the initialization parameters for Federation Web Services. This file is located in the one of the following directories:

   - *web_agent_home*/affwebservices/WEB-INF/classes
   - *sps_home*/secure-proxy/Tomcat/webapps/affwebservices/WEB-INF/classes

   web_agent_home

   Represents the installed location of the Web Agent

   sps_home

   Represents the installed location of the SPS federation gateway

3. Enable error and trace logging for the Federation Web Services application. Enable logging in the LoggerConfig.properties file. The logs enable you to see the communication between the asserting party and the relying party.

   - Error logging is recorded in the affwebserv.log file, the default error log file.
   - Trace logging is recorded in the FWSTrace.log, the default trace log file.

4. Test Federation Web Services by opening a web browser and entering the following link:

   http://*fqhn*:*port_number*/affwebservices/assertionretriever

   fqhn

   Defines the fully qualified host name.

   port_number

   Defines the port number of the server where the Federation Web Services application is installed.

   For example:

   http://myhost.ca.com:81/affwebservices/assertionretriever

   If Federation Web Services is operating correctly, you see the following message:

   ```
   Assertion Retrieval Service has been successfully initialized.
   The requested servlet accepts only HTTP POST requests.
   ```

   This message indicates that Federation Web Services is listening for data activity. If Federation Web Services is not operating correctly, you receive a message that the Assertion Retrieval Service has failed. If the test fails, look at the Federation Web Services log.

Allow Access to Federation Web Services (asserting party)

When you install the Policy Server, CA SiteMinder® creates policies for the Federation Web Services (FWS) application. The FWS application is installed with the Web Agent Option Pack. For a few federation features, the relying party needs permission to access the protected FWS service. Adding a relying partner to a policy is a task you do only at the asserting party.

For example, for HTTP-Artifact binding for single sign-on, a policy protects the service from which CA SiteMinder® retrieves an assertion. For CA SiteMinder® to retrieve the assertion for a specific relying partner, that partner must be added as a user to the policy that protects the service.

Grant access to specific FWS policies (see page 113) that apply to features configured for your federation partnership.

Enable the Signing of SAML Post Responses

Signing SAML POST responses is a SAML specification requirement. To sign SAML POST responses, add a private key and certificate to the certificate data store at the asserting party.

For instructions on importing keys and certificates into the data store, see the *Policy Server Configuration Guide*.

Create Links to Target Resources (optional)

Go to one of the following:

- Links for SAML 1.x Single Sign-On (see page 80)
- Links for SAML 2.0 Single Sign-On at the Identity Provider (see page 82)
- Links to Initiate WS-Federation Single Sign-on (see page 82)

Initiate SAML 1.x Single Sign-On at the Producer

At the SAML 1.x producer, create pages that contain links which direct the user to the consumer site. Each link represents an intersite transfer URL. The user has to visit the intersite transfer URL, which sends a request to the producer-side Web Agent. The user is then redirected to a consumer site.

The link that the user selects at the producer must contain certain query parameters. These parameters are part of an HTTP GET request to the producer Web Agent.

For the SAML artifact profile, the syntax for the intersite transfer URL is:

```
http://producer_site/affwebservices/public/intersitetransfer?SMASSERTIONREF=
QUERY&NAME=affiliate_name&TARGET=http://consumer_site/target_url?query_parame
ter_name%3Dquery_parameter_value%26query_parameter_name%3Dquery_parameter_val
ue&SMCONSUMERURL=http://consumer_site/affwebservices/public/samlcc&AUTHREQUIR
EMENT=2
```

producer_site

Specifies the server and port number of the system hosting the Web Agent Option Pack or the SPS federation gateway, depending on which components are installed in your federation network.

consumer_site

Specifies the server and port number of the system hosting the Web Agent Option Pack or the SPS federation gateway, depending on which components are installed in your federation network.

For the SAML POST profile, the syntax for the intersite transfer URL is:

```
http://producer_site/affwebservices/public/intersitetransfer?SMASSERTIONREF=
QUERY&NAME=affiliate_name&TARGET=http://consumer_site/target_url
```

producer_site

Specifies the server and port number of the system hosting the Web Agent Option Pack or the SPS federation gateway, depending on which components are installed in your federation network.

consumer_site

Specifies the server and port number of the system hosting the Web Agent Option Pack or the SPS federation gateway, depending on which components are installed in your federation network.

Note: The SAML POST profile does not use SMCONSUMERURL and AUTHREQUIREMENT parameters. However, if you include one of these parameters in the intersite transfer URL, include the other parameter.

More Information:

Create Links to Consumer Resources (SAML 1.x) (see page 144)

Initiate SAML 2.0 Single Sign-On at the Identity Provider

If a user visits the Identity Provider before going to the Service Provider (POST or artifact binding), initiate an unsolicited response at the Identity Provider. To initiate an unsolicited response, the Federation Web Service application and assertion generator accept an HTTP Get request with a query parameter. This query parameter indicates the Service Provider ID for which the IdP generates the response.

For SAML 2.0 artifact or post profile, the syntax for the link is:

http://*IdP_server:port*/affwebservices/public/saml2sso?SPID=*SP_ID*

idp_server:port

Identifies the web server and port hosting the Web Agent Option Pack or SPS federation gateway.

SP_ID

Service Provider ID value. The entity ID is case-sensitive. Enter it exactly as it appears in the Administrative UI.

Add the ProtocolBinding query parameter (see page 212) to this link depending on which bindings are enabled.

Note: You do not need to HTTP-encode the query parameters.

You can also initiate single sign-on at the Service Provider.

More information:

Unsolicited Response Query Parameters that the IdP Uses (see page 212)

Initiate WS-Federation Single Sign-on at the Account Partner

To initiate WS-Federation single sign-on, a user clicks on a page with a hard-coded HTML link. This HTML link directs the browser of the user to the single sign-on service at the Account Partner. The Account Partner then redirects the user to the Resource Partner.

The link that initiates single sign-on can be included at any site, but it must always first direct the user to the Account Partner.

The syntax for the link is:

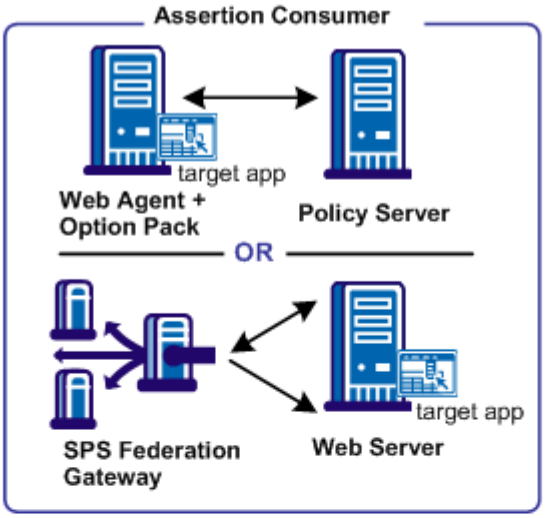
https://*AP:port*/affwebservices/public/wsfedsso?wa=wsignin1.0&wtrealm=*RP_ID*

ap_server:port

Specifies the server and port number of the system at the Account Partner. The system is hosting the Web Agent Option Pack or the SPS federation gateway, depending on which component is installed in your federation network.

Note: You do not need to HTTP-encode the query parameters.

Set Up Relying Party Components



1. **Install Policy Server**
   Policy Server Installation Guide
2. **Configure the SAML authentication scheme for each producer/IdP/AP**
3. **Create realms, rules, and policies to protect target resources.**
4. **Install and configure a Web Agent or an SPS Federation Gateway**
   Web Agent Installation Guide or SPS Administration Guide (Skip steps 6 and 7 if using the SPS)
5. **Install a web or application server for Federation Web Services**
6. **Install the Web Agent Option Pack**
   Web Agent Option Pack Guide
7. **Configure Federation Web Services**
8. **Protect Federation Web Services**
9. **For artifact SSO, set up certificate data store**

Except where noted, see this guide for configuration instructions.

Many of the steps for setting up a Policy Server and Web Agent at the relying party are similar to the steps for the asserting party, with the following exceptions:

- you do not configure consumers, Service Providers, or Resource Partners
- you configure a SAML or WS-Federation authentication scheme at the Policy Server

The following illustration shows the required tasks for the SAML 1.x Consumer, the SAML 2.0 Service Provider, or the WS-Federation Resource Partner.

Note: This procedure assumes that the target resources exist at the relying party website.

Install the Relying Party Policy Server

Install the Policy Server at the relying party site. The Policy Server provides functions such as the federation authentication schemes and the Assertion Generator.

For more information, see the *Policy Server Installation Guide* and the *Policy Server Configuration Guide.*

At the relying party, do the following:

1. Install the Policy Server.
2. Set up a policy store.

   Important! If you initialize a new policy store, the Policy Server installer automatically imports the affiliate objects in the ampolicy.smdif file. These objects are necessary for federation. If you use an existing policy store, import the affiliate objects manually. To verify that the import is successful, log in to the Administrative UI and navigate to Policy, Domain, Domains. If the import is successful, you can see the FederationWebServices domain object in the list.
3. Set up a user store and add users permitted to access target resources.

Configure a SAML or WS-Federation Authentication Scheme

At the relying party Policy Server, configure an authentication scheme (artifact, POST profile, SAML 2.0, WS-Federation) for each asserting party.

Important! The name of the partner that you specify for the authentication scheme must match the name of the relying party that you specify at the asserting party.

Specifically:

- For SAML 1.x authentication schemes, the Affiliate Name field of the scheme configuration must match an Affiliate Name for an affiliate object at the producer site.
- For SAML 2.0, the equivalent field is the SP ID, which must match the SP ID at the Identity Provider.
- For WS-Federation, the Resource Partner ID for the scheme configuration must match the Resource Partner ID at the Account Partner.

More Information:

Configure as a SAML 1.x Consumer (see page 147)
Configure a SAML 2.0 Service Provider (see page 231)
Configure CA SiteMinder® as a WS-Federation Resource Partner (see page 295)

Protect Target Resources at the Relying Party

After creating a SAML or WS-Federation authentication scheme, assign the scheme to a unique realm or a single custom realm. The realm is the collection of target resources at the relying party that require an assertion for user access. The relying party identifies target resources in one of the following ways:

- TARGET variable in the intersite transfer URLs (SAML 1.x).
- AuthnRequest URL (SAML 2.0 and WS-Federation).
- Authentication scheme configuration (SAML 2.0 and WS-Federation).

After you create a realm and assign a SAML or WS-Federation authentication scheme to it, create a rule for the realm, then add the rule to a policy that protects the resource.

Install a Web Agent or SPS Federation Gateway (Relying Party)

The Web Agent is a required component in a CA SiteMinder® legacy federation network. You can either install a Web Agent on a web server or install an SPS federation gateway, which has an embedded web agent.

At the relying party, set up the following components:

1. Install one of the following components:
   - Web Agent

     For instructions, see the *Web Agent Installation Guide.*
   - SPS federation gateway

     For instructions, see the *Secure Proxy Server Administration Guide*.
2. Configure the Web Agent or SPS federation gateway.

Install a Web or Application Server for the Web Agent Option Pack (Relying Party)

If you are implementing legacy federation with a Web Agent and Web Agent Option Pack (not with an SPS federation gateway), install the Web Agent Option Pack. Install this component on a web or application server.

At the relying party:

1. Install one of the following servers to run Federation Web Services, the application that is installed with the Web Agent Option Pack.
   - Web server running ServletExec
   - WebLogic Application Server
   - WebSphere Application Server
   - JBOSS Application Server
   - Tomcat Application Server
2. Deploy Federation Web Services on these systems.

Install the Web Agent Option Pack at the Relying Party

The Web Agent Option Pack supplies the Federation Web Services application, which is a required component for legacy federation.

At the relying party:

1. Install the Web Agent Option Pack.

   For instructions, see the *Web Agent Option Pack Guide*.

2. Verify that you install a JDK. The Web Agent Option Pack requires this JDK.

   To determine the required JDK version, go to the Technical Support site and search for CA SiteMinder® Platform Matrix.

Note: The SPS federation gateway can replace the Web Agent and Web Agent Option Pack to provide the Federation Web Services application functions. For information about installing and configuring the SPS federation gateway, see the *Secure Proxy Server Administration Guide*.

Configure Federation Web Services at the Relying Party

These steps enable you to set up the Federation Web Services application. The Federation Web Services application is installed on the server with the Web Agent Option Pack or the SPS federation gateway.

To configure Federation Web Services at the relying party

1. Configure one of the supported application servers to use the Web Agent Option Pack. Refer to the Web Agent Option Pack deployment instructions.

   If you are using the SPS federation gateway, the Federation Web Services application is already deployed.

2. Set the AgentConfigLocation parameter in the AffWebServices.properties file to the full path to the WebAgent.conf file. Verify that the syntax is correct and the path appears on one line in the file.

   The AffWebServices.properties file contains the initialization parameters for Federation Web Services. This file is located in the one of the following directories:

   - *web_agent_home*/affwebservices/WEB-INF/classes
   - *sps_home*/secure-proxy/Tomcat/webapps/affwebservices/WEB-INF/classes

   web_agent_home

   Represents the installed location of the Web Agent

   sps_home

   Represents the installed location of the SPS federation gateway

3. Enable error and trace logging for Federation Web Services application. Logging is enabled in the LoggerConfig.properties file. The logs enable you to see the communication between the asserting party and the relying party.
   - Error logging is recorded in the affwebserv.log file, the default error log file.
   - Trace logging is recorded in the FWSTrace.log, the default trace log file.
4. Test Federation Web Services by opening a web browser and entering the following link:

   http://*fqhn*:*port_number*/affwebservices/assertionretriever

   fqhn

   Defines the fully qualified host name.

   port_number

   Defines the port number of the server where the Federation Web Services application is installed.

   For example:

   http://myhost.ca.com:81/affwebservices/assertionretriever

   If Federation Web Services is operating correctly, the following message appears:

   ```
   Assertion Retrieval Service has been successfully initialized.
   The requested servlet accepts only HTTP POST requests.
   ```

   This message indicates that Federation Web Services is listening for data activity. If Federation Web Services is not operating correctly, you see a message that the Assertion Retrieval Service has failed. If the test fails, look at the Federation Web Services log.

More Information:

Configure Federation Web Services (Asserting Party) (see page 78)

Allow Access to Federation Web Services (asserting party)

When you install the Policy Server, CA SiteMinder® creates policies for the Federation Web Services (FWS) application. The FWS application is installed with the Web Agent Option Pack. For a few federation features, the relying party needs permission to access the protected FWS service. Adding a relying partner to a policy is a task you do only at the asserting party.

For example, for HTTP-Artifact binding for single sign-on, a policy protects the service from which CA SiteMinder® retrieves an assertion. For CA SiteMinder® to retrieve the assertion for a specific relying partner, that partner must be added as a user to the policy that protects the service.

Grant access to specific FWS policies (see page 113) that apply to features configured for your federation partnership.

Modify the Certificate Data Store for Artifact Single Sign-on (optional)

The certificate data store holds keys and certificates for PKI operations, such as encryption, decryption, signing, verification and client authentication.

If you are implementing artifact single sign-on, the certificate data store at the asserting party holds the certificate authority certificate for establishing an SSL connection. This SSL connection is between the relying party and the asserting party. This SSL connection secures the back channel that the assertion is sent across for artifact single sign-on.

A set of common root CAs is shipped in the certificate data store. To use root CAs for web servers that are *not* in the data store, import these root CAs.

For detailed information about the certificate data store, see the *Policy Server Configuration Guide*.

Create Links to Initiate Single Sign-on (optional)

For SAML 2.0 and WS-Federation, if a user visits the relying party before visiting the asserting party, establish hard-coded links. The hard-coded links redirect the user to the asserting party to fetch the authentication context. This authentication context consists of the characteristics that enable the relying party to understand how the user was authenticated.

More Information:

Initiate SAML 2.0 Single Sign-on at the SP (optional) (see page 90)
Initiate WS-Federation Single Sign-on at the Resource Partner (see page 90)

Initiate SAML 2.0 Single Sign-on at the SP (optional)

If a user visits the Service Provider before visiting the Identity Provider, the Service Provider must redirect the user to the Identity Provider. At the Service Provider, create an HTML page that contains hard-coded links to the AuthnRequest Service. The AuthnRequest service, in turn, redirects the user to the Identity Provider to fetch the authentication context.

Note: The HTML page has to reside in an unprotected realm.

The hard-coded link that the user clicks at the Service Provider must contain certain query parameters. These parameters become part of an HTTP GET request to the AuthnRequest service. The AuthnRequest service is on the Policy Server at the Service Provider.

For SAML 2.0 (artifact or profile), the syntax for the link is:

http://*sp_server:port*/affwebservices/public/saml2authnrequest?ProviderID=*IdP_ID*

sp_server:port

Specifies the server and port number of the Service Provider hosting the Web Agent Option Pack or the SPS federation gateway.

IdP_ID

Specifies the Identity Provider ID.

You can add the ProtocolBinding query parameter to this link depending on which bindings are enabled. For more information about configuring links at the Service Provider, see Set Up Links at the IdP or SP to Initiate Single Sign-on.

Note: You do not need to HTTP-encode the query parameters.

You can also create links at the Identity Provider.

Initiate WS-Federation Single Sign-on at the Resource Partner

If a user visits the Resource Partner before visiting the Account Partner, the Resource Partner must redirect the user to the Account Partner. Create an HTML page, such as a site selection page that contains links to Account Partners with which to authenticate. Upon selecting a link, the user is directed to the single sign-on service at the Account Partner.

Note: The site selection page has to reside in an unprotected realm.

The hard-coded link that the user clicks at the Resource Partner must contain certain query parameters. These parameters are part of an HTTP GET request to the Single Sign-on Service at the Policy Server of the Account Partner.

The syntax for the link is:

https://*host:port*/affwebservices/public/wsfedsso?wa=wsignin1.0&wtrealm=*RP_ID*

host:port

Indicates the server and port number where the single sign-on service resides

RP_ID

Specifies the Resource Partner identity

Note: You do not need to HTTP-encode the query parameters.

Chapter 4: Configure the SAML 1.x Assertion Generator File

The Policy Server at the producer includes a component named the assertion generator. For SAML 1.x only, the AMAssertionGenerator.properties file is required for the assertion generator to generate assertions. This properties file also contains commented instructions, which you can read for more information about the settings in the file.

The installed location of this file is:

policy_server_home/config/properties

The assertion generator works without modifying the settings in this file. However, the file contains default values that are used in the assertions, so change these values for your network.

Updates to the AmAssertionGenerator.properties file are picked up after the Policy Server is restarted.

To configure the AMAssertionGenerator.properties file

1. Go to the directory *policy_server_home*/config/properties.
2. Open the AMAssertionGenerator.properties file in a text editor.
3. Modify the following parameters:

   AssertionIssuerID

   Specifies the URL that identifies the site issuing the assertion.

   This URL must be the same value as the Issuer field that you complete for a SAML authentication scheme.

   SecurityDomain

   Identifies the domain of the producer, such as example.com.

   SourceID

   Specifies for the SAML 1.x artifact profile only, a unique ID in the artifact that identifies the producer. For more information, see the SAML specification at the OASIS website.

Important! The values in this file must match the values for the equivalent settings at the consumer site.

Chapter 5: Review the JVMOptions File for the JVM

The JVMOptions.txt file contains the settings that the Policy Server uses when creating the Java virtual machine that is used to support Federation Web Services. SAML 1.x, SAML 2.0, and WS-Federation use this file.

During a Policy Server upgrade, the existing JVMOptions.txt file is renamed to JVMOptions.txt.backup. A new JVMOptions.txt file is created. If the original file included customized parameters, modify the newly created file to include these customized parameters.

The installed location of this file is:

policy_server_home/config/

Important! If you update the JVMOptions.txt file, restart the Policy Server for the changes to take effect.

Notes:

- In some environments, logging off a system while the Policy Server is running causes the Policy Server service to fail. The failure is the result of a JVM issue. To prevent the failure, add the -Xrs command to its own command line in the JVMOptions.txt file. This java command reduces usage of operating system signals by the Java virtual machine.

  This command is case-sensitive so be sure to capitalize the X.

- If you encounter errors relating to missing classes, modify the classpath directive in this file. For complete information about the settings contained in the JVMOptions.txt file, see your Java documentation. The Java compiler directive java.endorsed.dirs is used in the JVMOptions.txt file to control class loading.

Chapter 6: Storing User Session, Assertion, and Expiry Data

This section contains the following topics:

Federation Data Stored in the Session Store

The session store stores data for the following federation features:

- HTTP-Artifact single sign-on (SAML 1.x or 2.x)

  A SAML assertion and the associated artifact are generated at the asserting party. The artifact identifies the generated assertion. The asserting party returns the artifact to the relying party. The relying party uses the artifact to retrieve the assertion, which the asserting party stores in the session store.

  A persistent session is required for this process to work.

  The SAML POST profile does not store assertions in the session store.

- HTTP-POST single use policy (SAML 2.0 and WS-Federation)

  The single use policy feature prevents assertions (POST binding) from being reused at the relying party to establish a second session. The relying party stores time-based data about the assertion, which is known as expiry data, in its session store. Expiry data helps ensure that the assertion is only used one time.

  A session store is required at the relying party, but a persistent session is not required.

- Authentication Session Variables Persistence (SAML 1.x and SAML 2.0)

  You can select the option Persist Authentication Session Variables when configuring federation at a relying party. This option instructs the Policy Server to save authentication context data in the session store as session variables. The Policy Server has access to these variables for use in authentication decisions.

- Assertion Attributes Persistence (all profiles)

  You can select Persist Attributes as a redirect mode at the relying party. The redirect mode determines how a user is redirected to the target application. The Persist Attributes mode instructs the Policy Server to store attributes that are extracted from an assertion in the session store. The attributes can then be supplied as HTTP header variables.

- Single logout (SAML 2.0)

  If single logout is enabled, either partner can stores information about the user session. The session information is kept in the session store. When a single logout request is completed, the session information for the user is removed, invalidating the session.

  A persistent session is required at the Identity Provider and Service Provider.

- Sign-out (WS-Federation)

  If sign-out is enabled, user context information is placed in the session store. This information enables the software to generate a sign-out request. When a sign-out request is completed, the session information for the user is removed, invalidating the user session.

  A persistent session is required at the Account Partner and Resource Partner.

Enable the Session Store

Enable the session store to store data when using SAML artifact single sign-on, single logout, WS-Federation sign-out, and a single use policy.

The session server database is where the Policy Server Session Server stores persistent session data.

Enable the session store from the Policy Server Management Console.

Follow these steps:

1. Log in to the Policy Server Management Console.
2. Select the Data tab.
3. Select Session Store from the drop-down list in the Database field.
4. Select an available storage type from the drop-down list in the Storage field.
5. Select the Session Store enabled check box.

   If you are going to use persistent sessions in one or more realms, enable the Session Server. When enabled, the Session Server impacts Policy Server performance.

   Note: For performance reasons, the session server cannot be run on the same database as the policy store. Therefore, the option to use the policy store database is disabled.
6. Specify Data Source Information appropriate for the chosen storage type.
7. Click OK to save the settings and exit the Console.
8. Stop and restart the Policy Server.

Environments that Require a Shared Session Store

The following CA SiteMinder® features require a shared session store to store SAML assertions and user session information.

To implement these features across a clustered Policy Server environment, set up the environment as follows:

- Configure the login realm for persistent sessions for all features *except* for an HTTP-POST single use policy.

  Persistent sessions are part of the realm configuration.

- For HTTP-Artifact single sign-on, share the session store at the Producer/Identity Provider site across all Policy Servers in the cluster.

  Sharing the session store verifies that all Policy Servers have access to assertions when each one receives a request for an assertion.

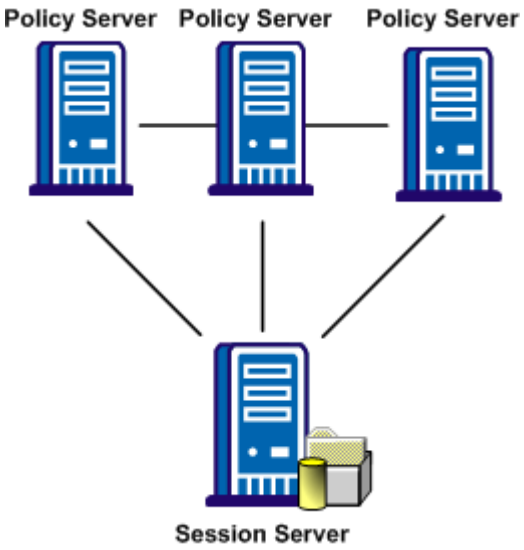
- For SAML 2.0 single logout and WS-Federation signout, share the session store at the asserting and relying party across all Policy Servers in the cluster.

  Sharing the session store verifies that all Policy Servers have access to user session data when each one receives a request for a session logout.

- For the HTTP-POST and WS-Federation single use policy feature, share the session store at the relying party across all Policy Servers in the cluster.

All Policy Servers that generate or consume assertions or process a persistent SMSESSION cookie must be able to contact the common session store. For example, a user logs in to example.com and gets a persistent session cookie for that domain. Every Policy Server that is handling requests for example.com must be able to verify that the session is still valid.

The following illustration shows a Policy Server cluster communicating with one session store:



To share a session store, use one of the following methods:

- Point all Policy Servers to one session store

  In the Policy Server Management Console, configure the Policy Server to use the designated session store.

- Replicate the session store across many session stores.

  For instructions on replicating a database, use the documentation for your database.

Chapter 7: Securing a Federated Environment

This section contains the following topics:

Protecting Federated Communication

Several mechanisms help secure transactions between federated partners, such as encrypting assertions and using SSL connections between partner sites.

When setting up a federated environment with CA SiteMinder®, here are some recommendations for protecting your environment:

- Enforcing the one time use of assertions.
- Securing connections at the asserting and relying parties.
- Protecting against cross-site scripting.

These topics are described in the sections that follow.

Setting a One Time Use Condition for an Assertion

In compliance with the SAML 1.x and 2.0 specifications, CA SiteMinder® can enforce the one time use of an assertion. By generating an assertion that is intended for one-time use, it tells the relying party not to retain the assertion for future transactions. Reusing an assertion beyond its validity results in authentication decisions that are based on out-of-date identity information.

If CA SiteMinder® is acting as the asserting party (Producer/IdP), you can configure the one time use of an assertion. For a SAML 1.x affiliate, you can select the **Set DoNotCache Condition** setting. For a SAML 2.0 IdP, you can select the **Set OneTimeUse Condition** setting. Both of these configuration settings enable CA SiteMinder® to insert the proper elements in an assertion that indicate the one-time use condition.

Note: Do not confuse the one time use of an assertion with the single use policy for SAML 1.x and 2.0 HTTP-POST single sign-on. The single use policy is only for POST transactions, but the one time use feature is for HTTP-Artifact and HTTP-POST.

Securing Connections Across the Federated Environment

Identity information that is sent between federated partners or a partner and an application is best protected when communication takes place over a secure connection.

Securing the Connection Between the Relying Party and the Target Application

Secure data transmission from the relying party to the client-site target application. Using a secure connection as the communication channel makes your environment less vulnerable to security attacks.

For example, an assertion can contain attributes that the relying party extracts and sends to the client application. The relying party can pass these attributes to the application using HTTP header variables or cookies. Attributes stored in headers or cookies can be overwritten at the client side, allowing a malicious user to impersonate other users. Using an SSL connection protects an environment from this type of security breach.

As a best practice, protect against this vulnerability by setting the UseSecureCookies parameter in the appropriate Agent Configuration Object (ACO). The UseSecureCookies parameter instructs Federation Web Services to generate cookies that are marked with the "secure" flag. This flag indicates that the cookie is sent only over an SSL communication channel.

Note: The ACO to modify differs depending on the setup of your federation environment. If you deploy Federation Web Services on the same system as the Web Agent is installed, edit the ACO for the Web Agent. If you deploy Federation Web Services on a different system than the Web Agent, edit the unique ACO you created for Federation Web Services.

Securing the Initial Authentication at the CA SiteMinder® Asserting Party

The initial authentication of a user at a CA SiteMinder® asserting party presents a potential vulnerability. When a user first authenticates to establish a user session at the asserting party, a session ID cookie is written to the browser. If the cookie is sent over a non-SSL connection, an attacker can obtain the cookie and can steal sensitive user information. The attacker can then use the information, for impersonation or identity theft.

As a best practice, protect against this vulnerability by setting the Web Agent parameter UseSecureCookies, which you can modify in the Agent Configuration Object. The UseSecureCookies parameter instructs the Web Agent to generate cookies that are marked with the "secure" flag. This flag indicates that the browser passes the cookie only over an SSL connection, which increases security. In general, establishing SSL connections for all URLs is recommended.

Protecting Against Cross-Site Scripting

A Cross Site Scripting (XSS) attack can occur when an application displays input text from a browser without filtering for characters that can form an executable script. The input text is typically data from a post or data from query parameters on a URL. The display of these characters in a browser can lead to an unwanted script being executed on the browser.

CA SiteMinder® provides several JSPs for use with CA SiteMinder® federation functionality. These JSPs check characters in a request to be sure that unsafe information in the output stream is not displayed in the browser.

When CA SiteMinder® receives a federation request, the following JSPs scan the decoded values for cross-site scripting characters:

- idpdiscovery.jsp

  Used at the relying party for Identity Provider Discovery.

- linkaccount.jsp

  Used at the relying party for dynamic account linking.

- sample_application.jsp

  Used at the IDP to initiate single sign-on. You can use this sample application to direct the user to the SSO Service and then to the custom web application. Typically, you use your own application.

- signoutconfirmurl.jsp

  Used at the Account Partner for WS-Federation sign out.

- unsolicited_application.jsp

  Used for IdP-initiated single sign-on when the user is sent directly to the web application and not initially to the SSO Service.

The pages scan the request for the following characters:

Character	Description
<	left angle bracket
>	right angle bracket
‘	single quotation mark
“	double quotation mark
%	percent sign
;	semi-colon
(	open (left) parenthesis

Character	Description
)	closed (right) parenthesis
&	ampersand
+	plus sign

Each CA SiteMinder®-provided JSP contains a variable that defines the characters to scan. You can modify these JSPs to expand the character set.

Chapter 8: Key and Certificate Management for Federation

Securing an assertion and encrypting data within the assertion is a critical part of partnership configuration. In a federation environment, key/certificate pairs and standalone certificates serve a number of functions:

- Signing/verification of assertions (all three profiles)
- Signing/verification of authentication requests (SAML 2.0 only)
- Signing/verification of single logout requests and responses (SAML 2.0)
- Signing back channel requests and responses for HTTP-Artifact SSO (SAML 1.1 and 2.0)
- Encryption/decryption of an entire assertion or part of an assertion (SAML 2.0)
- Client credentials across the back channel for artifact single sign–on (SAML 1.1 and 2.0)

The *Policy Server Configuration Guide* contains information and instructions about managing keys and certificates.

You can use SSL server certificates to do the following tasks:

- Manage federation traffic across an SSL connection.
- Secure communication across the back channel for artifact single sign-on.

Refer to instructions for enabling SSL for the web server where the CA SiteMinder® Web Agent is installed.

Note: If you enable SSL, it affects all URLs for all services, even the Base URL parameter. This means that all service URLs must begin with https://.

SAML 2.0 Signing Algorithms

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.

Chapter 9: User Directory Configuration for Federation

Directory connections resolve how CA SiteMinder® establishes a context for user identities. The software uses these connections to verify user identities and to retrieve the user attributes that are a part of the user record.

The asserting party determines which users it can create assertions for by authenticating each user against a user directory. At the relying party, when the assertion is presented during authentication, the relying party looks in the user directory for the user record.

Configure user directories before you select users for federated transactions. To configure user directories, see the *Policy Server Configuration Guide.*

Note: If you plan 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.

Chapter 10: Creating Affiliate Domains

This section contains the following topics:

Affiliate Domain Overview

An affiliate domain is a logical grouping of federated entities that are associated with one or more user directories.

The affiliate domain not only contains federated entities but it also defines which user directories are associated with the domain. To generate an assertion, CA SiteMinder® as an Identity Provider must have access to the user directory where a user record is defined. The Policy Server locates a user record by querying the user directories specified in the search order of the affiliate domain.

The search order is defined when you add user directory connections to an affiliate domain. You have the option of shifting the order of directories.

Affiliate domains require one or more administrator accounts that can modify the objects in the domain. System-level administrators can manage all objects in any domain; they have the permission Manage Affiliates. A system administrator that can grant control over a policy domain to other administrators has the permission Manage System and Domain Objects.

Configure an Affiliate Domain

You can add a domain object, select users who have access to resources at the consumer, Service Provider, or Resource Partner, and add associated entities.

Follow these steps:

1. Log In to the Administrative UI.
2. Click Federation, Legacy Federation, Affiliate Domains.
3. Click Create Affiliate Domain.
4. In the General settings, enter a name and a brief description for the affiliate domain.
5. In the User Directories section, click Add/Remove.

6. Move the user directories that you want to associate with the domain from Available Members to Selected Members.

   Specify the directories that store the records of users who you want to permit access to the affiliate resources.

7. Click OK.

   The selected directories appear in the User Directories table.

   If there are no existing directories, create a user directory by clicking Create. When you complete the required information, the directory you created appears in the User Directories table.

8. Optionally, in the User Directories table, use the arrows on the right to adjust the order of directories the table. Use the arrows on the left of to edit the details of a directory.

   The order that the directories appear is the order in which CA SiteMinder® searches to find user records, starting from the top of the list.

9. Click Submit.

   The affiliate domain is created.

The next step is to add partners to the affiliate domain and configure CA SiteMinder® as the asserting party in the federated partnership.

More information:

Configure a SAML 1.x Producer (see page 117)
Configure a SAML 2.0 Service Provider (see page 231)
Configure CA SiteMinder® as a WS-Federation Resource Partner (see page 295)

Add Entities to an Affiliate Domain

Configure CA SiteMinder® to perform the role of the asserting party in the federated partnership. For CA SiteMinder® to act as the asserting party, add partners to an affiliate domain. When a partner sends an authentication request, CA SiteMinder® can generate an assertion in response.

You can add the following entities to an affiliate domain:

- SAML 1.x Affiliates
- SAML 2.0 Service Providers
- WS-Federation Resource Partners

Note: These entities must have permission to access Federation Web Services (see page 131) at the asserting party.

For instructions on adding partners to an affiliate domain, see one of the following sections:

- Configure CA SiteMinder® as a SAML 1.x Producer (see page 117).
- Configure CA SiteMinder® as a SAML 2.0 Identity Provider (see page 175).
- Configure CA SiteMinder® as an Account Partner. (see page 273)

More information:

Configure a SAML 1.x Producer (see page 117)
Configure a SAML 2.0 Identity Provider (see page 175)
Configure a WS-Federation Account Partner (see page 273)

Chapter 11: Grant Access to Federation Web Services

This section contains the following topics:

Policies that Protect Federation Web Services

When you install the Policy Server, CA SiteMinder® creates policies for several services. These services comprise the Federation Web Services (FWS) application. For a few federation features, the relying party needs permission to access the associated protected service.

Adding a relying partner to a policy is a task that is done only at the asserting party.

For example, for the HTTP-Artifact binding, a policy protects the service from which CA SiteMinder® retrieves an assertion. For CA SiteMinder® to retrieve the assertion for a specific relying partner, that partner must be added as a user to the policy that protects the service.

The following table lists the FWS policy objects that are related to FWS services.

Object Type	Object Name
Domain	FederationWebServicesDomain
Realm	FederationWebServicesRealm public
Agent Group	FederationWebServicesAgentGroup
Rule	SAML2FWSAttributeServiceRule FederationWSSessionServiceRule SAML2FWSArtifactResolutionRule FederationWSAssertionRetrievalServiceRule FederationWSNotificationServiceRule

Object Type	Object Name
Policy	SAML2FWSArtifactResolutionServicePolicy SAML2FWSAttributeServicePolicy FederationWSAssertionRetrievalServicePolicy FederationWSNotificationServicePolicy FederationWSSessionServicePolicy
Variables	AllowNotification AllowSessionSync
User Directories	FederationWSCustomUserStore SAML2FederationCustomUserStore

Features Associated with FWS Policies

The policies that CA SiteMinder® creates support the following legacy federation features:

FWS Policy	Federation Feature
SAML2FWSArtifactResolutionServicePolicy	Protects the artifact resolution service for SAML 2.0 artifact single sign-on
FederationWSAssertionRetrievalServicePolicy	Protects the assertion retrieval service for SAML 1.x artifact single sign-on
SAML2FWSAttributeServicePolicy	Protects the attribute authority service for SAML 2.0
FederationWSNotificationServicePolicy	Protects the notification service. Notifications are only available if the SAML Affiliate Agent is the consumer.
FederationWSSessionServicePolicy	Protects the session service for session management. Session management is available only if the SAML Affiliate Agent is the consumer.

Enforce the Policies that Protect Federation Web Services

If you are implementing federation features with FWS policies, the relying party needs permission to access the protected service.

Granting access involves the following tasks:

- Adding the Web Agent that protects the FWS application to the agent group FederationWebServicesAgentGroup.
- Adding relying partners as users that are permitted to access the specific service.

  Other than adding users to a given policy, all other policy objects are set up automatically.

Detailed procedures for enforcing the HTTP-Artifact assertion retrieval and attribute authority policies are in the relevant sections for those features.

More information:

Grant Access to the Service for Assertion Retrieval (Artifact SSO) (see page 130)
Grant Relying Partners Access to the Attribute Authority Service (see page 331)

Chapter 12: Configure a SAML 1.x Producer

This section contains the following topics:

Prerequisites for an Asserting Partner(legacy)

To configure an asserting partner, verify the following conditions:

- The Policy Server is installed.
- One of the following options is installed:
  - The Web Agent and the Web Agent Option Pack. The Web Agent authenticates users and establishes a CA SiteMinder® session. The Option Pack provides the Federation Web Services application. Be sure to deploy the FWS application on the appropriate system in your network.
  - The SPS federation gateway has an embedded Web Agent and the Federation Web Services application on the embedded Tomcat web server.

  For more information, see the *Web Agent Option Pack Guid*e.
- Private keys and certificates are imported for functions that require signing and decrypting messages.
- A relying partner is set up within the federated network.

How To Configure a Producer

CA SiteMinder®, as an SAML producer generates assertions for its business partners, the consumers. To establish a federated partnership, the producer needs information about each partner, referred to as an affiliate in the Administrative UI. Create an affiliate object for each partner and define how the two entities communicate to pass assertions and to satisfy profiles, such as single sign-on.

The following configuration tasks at the producer are required:

1. Associate the affiliate with an affiliate domain.
2. Configure the general settings for the affiliat (see page 120)e.
3. Select the users for which the producer generates assertions (see page 123).
4. Configure an assertion (see page 126).
5. (HTTP-Artifact SSO only)
   a. Enable the session store to store assertions. Manage the session store using the Policy Server Management Console.
   b. Permit access to the assertion retrieval service (see page 131) for each applicable relying party.
6. Create links to initiate single sign-on (see page 144).
7. Complete optional configuration tasks (see page 118).

Tips:

- Certain parameter values at the Producer and Consumer are required to match for the configuration to work. A list of those parameters is in Configuration Settings that Must Use the Same Values (see page 361).
- Verify that you are using the correct URLs for the Federation Web Services servlets. The URLs are listed in Federation Web Services URLs Used in SiteMinder Configuration (see page 367).

Optional Configuration Tasks for Identifying an Affiliate

The following tasks are optional for identifying an affiliate.

- Configure attributes for inclusion in the assertions (see page 138).
- Configure time restrictions for affiliate operation (see page 122).
- Set IP address restrictions (see page 123) to limit the addresses that access the affiliate.
- Customize the content of an assertion (see page 142) using the Assertion Generator plug-in.

Navigating Legacy Federation Dialogs

The Administrative UI provides two ways to navigate to the legacy federation configuration dialogs.

You can navigate in one of two ways:

- Following a wizard to configure a new legacy federation object.

  When you create an object, a page displays with a configuration wizard. Follow the steps in the configuration wizard to create the object.

- Selecting tabs to modify an existing legacy federation object.

  When you modify an existing object, a page displays with a series of tabs. Modify the configuration from these tabs. These tabs are the same as the steps in the configuration wizard.

Associate a SAML 1.x Affiliate with an Affiliate Domain

An affiliate domain is a logical grouping of federation partners. You associate an affiliate with an affiliate domain so that CA SiteMinder® is able to recognize it.

Follow these steps:

1. Log in to the Administrative UI.
2. Click Federation, Legacy Federation, Affiliates.
3. Click Create Affiliate.
4. Select the affiliate domain where this affiliate belongs.
5. Click Next.

The affiliate is associated with an affiliate domain. The next step in is to provide some general information (see page 120) about the affiliate.

More Information:

Authenticate Users with No CA SiteMinder® Session (SAML 1.x) (see page 121)

Complete the General Settings for the Affiliate

Configure the general settings for the affiliate.

To provide general information about the affiliate

1. Begin at the General step in the configuration wizard.
2. Complete the following required fields in the General section.
   - Name
   - Password and Confirm Password
   - Authentication URL

     This URL must point to the redirect.jsp file -- for example:

     http://myserver.mysite.com/siteminderagent/redirectjsp/redirect.jsp

     myserver Identifies the web server with the Web Agent Option Pack or the SPS federation gateway.

     Be sure to create a policy to protect the Authentication URL.
3. Select Enabled to activate the affiliate object.
4. (Optional) Select Use Secure URL.

   The Use Secure URL feature instructs the SSO Service to encrypt the SMPORTALURL query parameter that it appends to the Authentication URL before redirecting the user to establish a CA SiteMinder® session. Encrypting the SMPORTALURL protects it from modification by a malicious user.

   Note: If you select this check box, set the Authentication URL field to the following URL:

   http(s)://*idp_server:port*/affwebservices/secure/secureredirect.

   Click Help for more details about this field.
5. (Optional) Complete the fields in the Restrictions and Advanced sections.
6. Click Next.

Authenticate Users with No CA SiteMinder® Session (SAML 1.x)

When you add a consumer to an affiliate domain, you are required to set the Authentication URL field. The Authentication URL must point to the redirect.jsp file. The purpose of this URL is to establish a session at the producer.

The redirect.jsp file is installed at the producer where you install the Web Agent Option Pack or the SPS federation gateway. Protect the redirect.jsp file with a CA SiteMinder® policy so that users who request a protected resource are asked to authenticate. The Web Agent presents the challenge because the user does not have a CA SiteMinder® session.

After a user is authenticated and successfully accesses the redirect.jsp file, a session is established. The redirect.jsp file redirects the user back to the producer Web Agent. The Agent can process the request and can generate the SAML assertion.

The procedure for protecting the Authentication URL is the same in all of the following set-ups:

- Web Agent Option Pack that is installed on the same system as the Web Agent.
- Application server with a Web Agent installed on a web server proxy.
- Application server installed with an Application Server Agent.
- SPS federation gateway that is installed at the asserting party.

Configure a Policy to Protect the Authentication URL

To protect the Authentication URL

1. Log in to the Administrative UI.
2. Create Web Agents to bind to the realms that you define for the asserting party web server. Assign unique agent names for the web server and the FWS application or use the same agent name for both.
3. Create a policy domain for the users who are challenged when they try to access a consumer resource.
4. Select the users that must have access to the resources that are part of the policy domain.
5. Define a realm for the policy domain with the following values:

   Agent

   Agent for the asserting party web server

   Resource Filter

   Web Agents r6.x QMR 6, r12.0 SP2, r12.0 SP3 and SPS federation gateway enter:

   /siteminderagent/redirectjsp/

The resource filter /siteminderagent/redirectjsp/ is an alias that the FWS application sets up automatically. The alias references include:

- Web Agent:

  web_agent_home/affwebservices/redirectjsp

- SPS federation gateway:

  sps_home/secure-proxy/Tomcat/webapps/affwebservices/redirectjsp

Persistent Session

For the SAML artifact profile only, select the Persistent check box in the Session section of the realm dialog. If you do not configure a persistent session, the user cannot access consumer resources.

For the remaining settings, accept the defaults or modify as needed.

6. Click OK to save the realm.
7. Create a rule for the realm. In the Resource field, accept the default value, the asterisk (*), to protect all resources for the realm.
8. Create a policy for the asserting party web server that includes the rule created in the previous step.
9. Complete the task Select Users for Which Assertions are Generated (see page 123).

Configure Time Restrictions for SAML 1.x Consumers (optional)

You can specify time restrictions that restrict when a consumer resource is available. When you specify a time restriction, access to the consumer resources is available only during the period specified. If a user tries accessing a resource outside of the allowed time period, the producer does not generate a SAML assertion.

Note: Time restrictions are based on the system clock of the server on which the Policy Server is installed.

To specify a time restriction

1. Begin at the General settings.

   In the Restrictions section of the page, click Set in the Time.

   The Time Restriction page displays.

2. Complete the schedule. This schedule grid is identical to the Time Restriction grid for rule objects. For more information, see the *Policy Server Configuration Guide*.
3. Click OK.

The time restriction schedule is set.

Configure IP Address Restrictions for SAML 1.x Consumers (optional)

You can specify an IP address, range addresses, or a subnet mask of the web server where the browser is running to access a consumer. If you specify IP addresses, the consumer only accepts users from the appropriate IP addresses.

To specify IP addresses

1. Begin at the General settings in the Administrative UI.

   In the Restrictions section of the page, click Add in the IP Address area.

   The IP Restrictions page appears.

2. Select the option for the type of IP address you are adding, then complete the associated fields for that address type.

   If you do not know the IP address but you know the domain name, click the DNS Lookup button. This button opens the DNS Lookup page. Enter a fully qualified host name in the Host Name field and click OK.

   The options are:

   - Single Host--specifies a single IP address that hosts the browser. If you specify a single IP address, users can access the consumer only from the specified IP address.
   - Host Name--specifies a web server using its host name. If you specify a host name, the consumer is only accessible to users from the specified host.
   - Subnet Mask--specifies a subnet mask for a web server. If you specify a subnet mask, the Service Provider is only accessible to users from the specified subnet mask. If you select this button, the Add An Address and Subnet Mask dialog opens. Use the Left and Right arrow buttons, or click and drag the slider bar to select a subnet mask.
   - Range--specifies IP address range. If you specify a range of IP addresses, the consumer only permits users from one of the IP addresses in the range of addresses. Enter a starting (FROM) and ending (TO) addresses to determine the range.

3. Click OK to save your configuration.

Select Users for Which Assertions are Generated

As part of the configuration at the asserting party, include a list of users and groups for which the Assertion Generator generates SAML assertions. The asserting party is either a SAML 1.x Producer, a SAML 2.0 Identity Provider, or a WS Federation Account Partner.

You can only add users and groups from directories that are in an affiliate domain.

To specify users and groups for federated transactions

1. Navigate to the Users settings for the partner you are configuring.

   The User Directories page displays entries for each user directory for the policy domain.

2. Add users or groups from the user directory to the policy.

   In each user directory table, you can select Add Members, Add Entry, Add All. Depending on which method you select, a dialog opens enabling you to add users.

   - 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.
   - If you select Add Entry, select users by manual entry (see page 125) in the User Directory Search Expression Edit dialog.

   Edit or delete a user or group by clicking the right arrow (>) or minus sign (-), respectively.

3. Select individual users, user groups, or both using whatever method and click OK.

   The User Directories page reopens and lists the new users in the user directory table.

More information:

Exclude a User or Group from Access to a Resource (see page 124)
Allow Nested Groups Access to Resources (see page 125)
Add Users by Manual Entry (see page 125)

Exclude a User or Group from Access to a Resource

You can exclude users or groups of users from obtaining an assertion.

Follow these steps:

1. Navigate to the User settings.
2. Select a user or group from the list for a particular user directory.
3. Click Exclude to exclude the selected user or group.

   The selection is reflected in the Administrative UI.

4. Click OK to save your changes.

Allow Nested Groups Access to Resources

LDAP user directories can contain groups that have subgroups. In complex directories, groups nesting in a hierarchy of other groups is one way to organize large amounts of user information.

If you enable a search for users in nested groups, any nested group is searched for the requested user record. If you do not enable nested groups, the Policy Server only searches the group you specify.

To enable searching in nested groups

1. Navigate to the Users settings.

   If the associated affiliate domain contains more than one user directory, each user directory appears in its own section.

2. Select the Allow Nested Groups check box to enable searching within nested groups.

Add Users by Manual Entry

When you specify users for assertion generation, one of the options is to identify users by manual entry.

Follow these steps:

1. Navigate to the Users settings for the partner you are configuring.

   If the affiliate domain contains more than one user directory, all the directories appear on the User Directories page.

2. Click Add Entry.

   The User Directory Search Express Edit page displays.

3. Select the search option then complete the fields for that search option.

   Where to Search

   For LDAP directories, select an option from the drop-down list:

   Validate DN

   LDAP search locates this DN in the directory.

   Search Users

   LDAP search is limited to matches in user entries.

Search Groups

LDAP search is limited to matches in group entries.

Search Organizations

LDAP search is limited to matches in organization entries.

Search Any Entry

LDAP search is limited to matches in user, group, and organization entries.

- For Microsoft SQL Server, Oracle and WinNT directories, you can enter a user name in the Manual Entry field.
- For a Microsoft SQL Server or Oracle, you can enter a SQL query instead. For example:

  SELECT NAME FROM EMPLOYEE WHERE JOB ='MGR';

  The Policy Server performs the query as the database user specified in the Username field of the Credentials and Connection tab for the user directory. When constructing the SQL statement for the Manual Entry field, be familiar with the database schema for the user directory. For example, if you are using the SmSampleUsers schema and you want to add specific users, select a user entry from the SmUser table.
- For an LDAP directory, enter **all** in the Manual Entry field to add all directory entries.

4. Click OK to save your changes.

Configure a SAML 1.x Assertion

At the producer site, determine how to deliver SAML assertions to a consumer. The assertion identifies the user to the consumer.

An assertion is an XML document that contains the following information:

- Information about the consumer
- Session information
- User attributes

For complete information about SAML assertions, refer to the SAML specification at the OASIS website.

To configure a SAML 1.x assertion

1. Navigate to the Assertions settings.
2. Complete the fields in the Assertion page. Click Help for the field descriptions.
   - Assertion Consumer URL (required for SAML POST; optional for SAML Artifact)

     For the SAML 1.x artifact binding, the Assertion Consumer URL takes precedence over the SMCONSUMERURL query parameter, which is a required intersite transfer URL parameter. The user selects this URL to initiate single sign-on. Malicious users can modify the query parameter and can send the user to an unauthorized site for artifact retrieval. To prevent the user from being misdirected, specify a value for the Assertion Consumer URL.
   - Skew Time Seconds

     Specifies the difference, in seconds, between the system clock at the Producer and the system clock at the Consumer. Skew Time is used for single sign-on and single logout.

     For single sign-on, the value of the Skew Time and the single sign-on validity duration determine how long an assertion is valid. Review how the assertion validity is calculated (see page 128) to understand more about the skew time.
3. Click Finish to save your selections.

The Assertions page also contains the optional Attributes (see page 138) section. This section lets you include attributes in the assertion.

More Information:

A Security Issue Regarding SAML 1.x Assertions (see page 127)

A Security Issue Regarding SAML 1.x Assertions

The SAML Assertion Generator creates an assertion that is based on a session for a user that has been authenticated at any authentication scheme protection level. You can control which users a producer generates assertions. You cannot control the protection level at which they are authenticated.

You can have resources that require a particular protection level. Your resources can be secured at different protection levels. Verify that when users authenticate they do so with the desired protection level.

Assertion Validity for Single Sign-on

For single sign-on, the values of the Skew Time and the Validity Duration determine how CA SiteMinder® calculates the total time that an assertion is valid. CA SiteMinder® applies the skew time to the generation and consumption of assertions.

Note: In this description, the asserting party is the SAML 1.x Producer, SAML 2.0 Identity Provider, or WS-Federation Account Partner. The relying party is the SAML 1.x Consumer, the SAML 2.0 Service Provider, or the WS-Federation Resource Partner.

In the assertion document, the NotBefore and NotOnOrAfter values represent the beginning and end of the validity interval.

At the asserting party, CA SiteMinder® sets the assertion validity. The validity interval is the system time when the assertion is generated. CA SiteMinder® sets the IssueInstant value in the assertion using this time then subtracts the skew time value from the IssueInstant value. The resulting time is the NotBefore value.

NotBefore=IssueInstant - Skew Time

To determine the end of the validity interval, CA SiteMinder® 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, CA SiteMinder® performs the same calculations as it does at the asserting party to determine if the assertion it receives is valid.

Calculating Assertion Validity with CA SiteMinder® at Both Sides of the Partnership

If CA SiteMinder® is at both sides of a partnership, the assertion validity is the sum of the validity duration plus two times the skew time. The equation is:

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

The initial part of the equation (2 x Skew Time + Validity Duration) represents the beginning and end of 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 legacy federation, the Validity Duration is only set at the asserting party.

Example

Asserting Party

The values at the asserting party are as follows:

- IssueInstant=5:00PM
- Validity Duration=60 seconds
- Skew Time = 60 seconds
- NotBefore = 4:59PM
- NotOnOrAfter=5:02PM

Relying Party

The relying party uses the NotBefore and NotOnOrAfter values from the assertion and applies its skew time to those values. This formula is how the relying party calculates new NotBefore and NotOnOrAfter values.

- Skew Time = 180 seconds (3 minutes)
- NotBefore = 4:56PM
- NotOnOrAfter=5:05PM

Assertion Validity Window

Using the values in this example, the calculation for the total assertion validity window is:

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

Configure an Assertion for One-time Use

In compliance with the SAML 1.x specification, CA SiteMinder® can enforce the one time use of an assertion. By generating an assertion for one-time use, the relying party knows not to retain the assertion for future transactions. Reusing an assertion beyond its validity results in authentication decisions using out-of-date identity information.

To configure an assertion for one time use

1. Navigate to the General settings for an Affiliate object.
2. In the Advanced section, select the Set DoNotCache Condition.
3. Click Submit.

Grant Access to the Service for Assertion Retrieval (Artifact SSO)

For HTTP-Artifact single sign-on, the relying party needs permission to access the policy that protects the FWS service for obtaining assertions.

To grant access:

- Add the Web Agent that protects the FWS application to the agent group FederationWebServicesAgentGroup.
- Add relying partners as users (see page 131) who are permitted to access the specific service.

  Other than adding users to a given policy, all other policy objects are set up automatically.

Add a Web Agent to the Federation Agent Group

Add the Web Agent that protects the FWS application to the Agent group FederationWebServicesAgentGroup.

- For ServletExec, this Agent is on the web server where the Web Agent Option Pack is installed.
- For an application server, such as WebLogic or JBOSS, this Web Agent is installed where the application server proxy is installed. The Web Agent Option Pack can be on a different system.

Follow these steps:

1. Log in to the Administrative UI.
2. Click Infrastructure, Agent, Agents.
3. Click Create Agent.
4. Specify the name of the Web Agent in your deployment. Click Submit.
5. Click Infrastructure, Agent, Agent Groups.
6. Select the FederationWebServicesAgentGroup entry.
7. Click Add/Remove and the Agent Group Members dialog opens.
8. Move the web agent from the Available Members list to the Selected Members list.
9. Click OK to return to the Agent Groups dialog.
10. Click Submit then click Close to return to the main page.

Add Relying Partners to the FWS Policy for Obtaining Assertions

If you are using HTTP-Artifact binding for single sign-on, the relying party in the partnership needs permission to access the assertion retrieval service. CA SiteMinder® protects the SAML 1.x and 2.0 retrieval services with a policy.

When you install the Policy Server, the FederationWebServicesDomain is installed by default. This domain includes the following policies for the service from which CA SiteMinder® retrieves assertions:

SAML 1.x

FederationWSAssertionRetrievalServicePolicy

SAML 2.0

SAML2FWSArtifactResolutionServicePolicy

Note: WS-Federation does not use the HTTP-Artifact profile. Therefore, this procedure does not apply to Resource Providers.

Grant access for these policies to any relevant relying partners.

Follow these steps:

1. In the Administrative UI, navigate to Policies, Domain, Domain Policies.

   A list of domain policies displays.

2. Select the policy for the SAML profile:

   SAML 1.x

   FederationWSAssertionRetrievalServicePolicy

   SAML 2.0

   SAML2FWSArtifactResolutionServicePolicy

   The Domain Policies page opens.

3. Click Modify to change the policy.

4. Select the Users tab.

5. In the dialog for the appropriate user directory, click Add Members:

   SAML 1.x

   FederationWSCustomUserStore

   SAML 2.0

   SAML2FederationCustomUserStore

   The User/Groups page opens.

   The affiliate domain that you previously configured is listed in the Users/Groups dialog. For example, if the affiliate domain is named fedpartners, the entry is **affiliate:fedpartners**.

6. Select the check box next to the affiliate domain with the partners that require access to the service. Click OK.

   You return to the User Directories list.

7. Click Submit.

   You return to the policies list.

Verify Basic Protection of the Assertion Retrieval Service

If you configure basic authentication to protect the assertion retrieval service, verify the protection.

Follow these steps:

1. Open a web browser.

   Access Federation Web Services by entering a fully qualified host name and port number for the server where the Federation Web Services application is installed. For example:

   SAML 1.x: http://idp-fws.ca.com:81/affwebservices/assertionretriever

   SAML 2.0: http://idp-fws.ca.com:81/affwebservices/saml2artifactresolution

   If the service is protected, CA SiteMinder® challenges you for credentials. Only an authorized affiliate is permitted access to Federation Web Services.

2. Enter a valid name and password that is for a relying partner that is configured at the Policy Server. The name and password are the credentials for the authentication challenge.

The authentication challenge indicates that the service is protected. If CA SiteMinder® does not present a challenge, the policy improperly configured.

Configure the Authentication Scheme that Protects the Artifact Service

For the HTTP-Artifact profile, the assertion retrieval service (SAML 1.x) and the artifact resolution service (SAML 2.0) retrieve the assertion at the asserting party. When these services send an assertion response to the relying party, they do so over a secure back channel. We strongly recommend that you protect these services and the communication across the back channel against unauthorized access.

Note: WS-Federation does not support the HTTP-Artifact profile.

To protect these services, specify an authentication scheme for the realm that contains the service at the asserting party. The authentication scheme dictates the type of credentials that the consuming service at the relying party must provide to access the relevant service across the back channel.

You can select one of the following authentication schemes:

- Basic (see page 134)
- Basic over SSL (see page 134)
- X.509 client certificate (see page 135)

Basic Authentication to Protect the Service that Retrieves Assertions

For HTTP-Artifact single sign-on, the asserting party sends the assertion across a secure back channel to the relying party. For basic authentication, configure a password to access to the service that resolves the artifact and retrieves the assertion. The service then sends the assertion across the back channel to the relying party.

You can use Basic authentication with SSL is enabled; however, SSL is not required.

Note: The password is only relevant if you use Basic or Basic over SSL as the authentication method across the back channel.

Follow these steps: for the SAML 1.x Assertion Retrieval Service

1. Log in to the Administrative UI.
2. Navigate to the General settings for the producer.
3. Enter a value for the following fields:
   - Password
   - Confirm Password
4. Click Submit to save the changes.

Follow these steps: for the SAML 2.0 Artifact Resolution Service

1. Log in to the Administrative UI.
2. Navigate to the Attribute settings for the Identity Provider.
3. In the Backchannel section, enter a value for the following fields:
   - Password
   - Confirm Password
4. Click Submit to save the changes.

Basic over SSL to Protect the Service that Retrieves Assertions

You can protect the assertion retrieval service (SAML 1.x) or the artifact resolution service (SAML 2.0) with a Basic over SSL authentication scheme. At the asserting party, a set of default policies to protect the service is already configured when you install the Policy Server.

The only configuration that is required is to enable SSL at each partner. No other configuration is required at the asserting or relying party. At the relying party, you can use one of the default root Certificate Authorities (CAs) in the certificate data store to establish an SSL connection. To use your own root CA instead of a default CA, import the CA certificate into the data store.

If you use Basic over SSL authentication scheme, all endpoint URLs have to use SSL communication. This means that the URLs must begin with **https://**. Endpoint URLs locate the various SAML services on a server, such as single sign-on, single logout, the Assertion Consumer Service, Artifact Resolution Service (SAML 2.0), and the Assertion Retrieval Service (SAML 1.x).

Client Certificate Auth to Protect the Service that Retrieves Assertion

You can protect the Assertion Retrieval Service (SAML 1.x) and the Artifact Resolution Service (SAML 2.0) with a client certificate authentication scheme. If the asserting party is configured to require client certificate authentication, the relying party makes a connection back to the asserting party and attempts to present a client certificate.

To use a client certificate authentication scheme:

1. Create a policy at the asserting party to protect the relevant service. This policy uses the client certificate authentication scheme.
2. Enable client certificate authentication for the back channel configuration at the relying party.
3. Enable SSL at each side of the partnership.

If you use Client Cert authentication, all endpoint URLs have to use SSL communication. Therefore, URLs must begin with **https://**. Endpoint URLs locate the various SAML services on a server, such as single sign-on, single logout, the Assertion Consumer Service, Artifact Resolution Service (SAML 2.0), and the Assertion Retrieval Service (SAML 1.x).

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.

Create the Policy to Protect the Retrieval Service

Create the policy at the asserting party to protect the service from which the asserting party retrieves the assertion.

Follow these steps:

1. For each affiliate requesting assertions, add a separate entry to a user directory. Create a user directory or use an existing directory.

   In the user record, enter the same value that is specified in the Name field of the affiliate general settings in the Administrative UI. For example, if Company A is the value of the Name field for the affiliate, the user directory entry is:

   uid=CompanyA, ou=Development,o=CA

   The Policy Server maps the subject DN value of the affiliate client certificate to this directory entry.

2. Add the configured user directory to the FederationWebServicesDomain.

3. Create a certificate mapping entry.

   Map the Attribute Name to the user directory entry for the affiliate. The attribute represents the subject DN entry in the certificate for the affiliate. For example, you select CN as the Attribute Name, and this value represents the affiliate named cn=CompanyA,ou=Development,o=partner.

   Navigate to Infrastructure, Directory, Certificate Mappings for the mapping settings.

4. Configure an X509 Client Certificate authentication scheme.

5. Create a realm under the FederationWebServicesDomain containing the following entries:

   Name

   any_name

   Example: cert assertion retrieval

   Agent

   FederationWebServicesAgentGroup

   Resource Filter

   /affwebservices/certassertionretriever (SAML 1.x)

   /affwebservices/saml2certartifactresolution (SAML 2.0)

   Authentication Scheme

   Client certificate authentication scheme created in the previous step.

6. Create a rule under the cert assertion retriever realm containing the following information:

   Name

   any_name

   Example: cert assertion retrieval rule

   Resource

   *

   Web Agent Actions

   GET, POST, PUT

7. Create a Web Agent response header under the FederationWebServicesDomain.

   The assertion retrieval service uses this HTTP header to verify that the affiliate is the site retrieving the assertion.

   Create a response with the following values:

   Name

   any_name

   Attribute

   WebAgent-HTTP-Header-Variable

   Attribute Kind

   User Attribute

   Variable Name

   consumer_name

   Attribute Name

   Enter the use directory attribute that contains the affiliate name value.

   Example: uid=CompanyA.

   Based on the following entries, the Web Agent returns a response named HTTP_CONSUMER_NAME.

8. Create a policy under the FederationWebServicesDomain containing the following values:

   Name

   any_name

   User

   Add the users from the user directory created in previously in this procedure.

Rule

rule_created_earlier_in_this_procedure

Response

response_created_earlier_in_this_procedure

The policy to protect the artifact resolution service is complete.

At the relying party, the administrator has to enable client certificate authentication across the back channel that connects to the relevant assertion service:

SAML 1.x: Enable client certificate authentication (see page 168) for the Assertion Retrieval Service

SAML 2.0: Enable client certificate authentication (see page 263) for the Artifact Resolution Service

Configure Attributes to Include in SAML 1.x Assertions (Optional)

You can include attributes in assertions. Servlets or applications can use attributes to display customized content for a user. User attributes, DN attributes, or static data can all be passed from the producer to the consumer in an assertion. When used with web applications, attributes can limit the activities of a user at the consumer. For example, the producer sends an attribute named Authorized Amount. The consumer sets this attribute to a maximum dollar amount that the user can spend.

Attributes take the form of name/value pairs and include information, such as a mailing address, business title, or an approved spending limit for transactions. When the consumer receives the assertion, it extracts the attributes. The consumer makes the attributes available to applications as HTTP header variables or HTTP cookie variables.

To pass the attributes, configure a response. The responses available for this purpose are:

- Affiliate-HTTP-Header-Variable
- Affiliate-HTTP-Cookie-Variable

The HTTP headers and HTTP cookies have size restrictions that assertion attributes cannot exceed. The size restrictions are as follows:

- For HTTP headers, CA SiteMinder® can send an attribute in a header up to the web server size limit for a header. Only one assertion attribute per header is allowed. See the documentation for your web server to determine the header size limit.
- For HTTP cookies, CA SiteMinder® can send a cookie up to the size limit for a cookie. Each assertion attribute is sent as its own cookie. The cookie size limit is browser-specific, and that limit is for all attributes being passed to the application, not for each attribute. See the documentation for your web browser to determine the cookie size limit.

More Information:

Configure Attributes for SAML 1.x Assertions (see page 139)
Use a Script to Create A New Response Attribute (see page 142)

Configure Attributes for SAML 1.x Assertions

You can configure responses to pass attributes from a SAML assertion to a target application at the consumer site.

To configure an attribute for an assertion

1. Navigate to the Assertions settings.
2. Click Add in the Attributes section.

   The Add Attribute dialog opens.
3. From the Attribute Type drop down, select whether you want to configure a header or cookie variable.
4. From the Attribute Setup section, select one of the following options in the Attribute Kind section:
   - Static
   - User Attribute
   - DN Attribute

   Click Help for the field descriptions.

   Your selection determines the available fields in the Attribute Fields section.

5. Complete the fields for the Attribute Kind you select. The Attribute Kind that you select determines which additional fields you must configure.

   Static

   Fill in the following fields:

   - Variable Name

     Enter the name for the attribute CA SiteMinder® returns to the affiliate.

   - Variable Value

     Enter the static text as the value for the name/value pair.

     For example, to return the name/value pair show_content=yes, enter show_content as the variable name and yes as the variable value.

   User Attribute

   Fill in the following fields:

   - Variable Name

     Enter the name for the attribute CA SiteMinder® returns to the consumer.

   - Attribute Name

     Enter the attribute in the user directory for the name/value pair.

     For example, to return the email address of a user to the consumer, enter email_address as the Variable Name, and email as the Attribute Name.

   DN Attribute

   Fill in the following fields:

   - Variable Name

     Enter the name for the attribute CA SiteMinder® returns to the consumer.

   - DN Spec

     Enter the distinguished name of the user group from which CA SiteMinder® retrieves the user attribute. The DN value is returned to the consumer. If you do not know the DN, click Lookup. Use the CA SiteMinder® User Lookup dialog to locate the user group and select a DN.

   - Attribute Name

     Enter the attribute in the user directory for this attribute for the name/value pair.

   If you selected Affiliate-HTTP-Cookie-Variable from the Attribute menu, the Variable Name field label changes to Cookie Name.

6. (Optional) To retrieve DN attributes from the nested groups, select the Allow Nested Groups check box in the Attribute Kind section.

7. Click OK to save your changes.

Specify the Maximum Length of Assertion Attributes

The maximum length for user assertion attributes is configurable. To modify the maximum length of assertion attributes, change the settings in the EntitlementGenerator.properties file.

The property name in the file is specific to the protocol you are configuring.

Follow these steps:

1. On the system where the Policy Server is installed, navigate to *policy_server_home*\config\properties\EntitlementGenerator.properties.
2. Open the file in a text editor.
3. Adjust the maximum user attribute length for the protocols in use in your environment. The settings for each protocol are as follows:

   WS-Federation

   Property Name:
   com.netegrity.assertiongenerator.wsfed.MaxUserAttributeLength

   Property Type: Positive Integer value

   Default Value: 1024

   Description: Indicates the maximum attribute length for WS-FED assertion attributes.

   SAML 1.x

   Property Name:
   com.netegrity.assertiongenerator.saml1.MaxUserAttributeLength

   Property Type: Positive Integer value

   Default Value: 1024

   Description: Indicates the maximum attribute length for SAML1.1 assertion attributes.

   SAML 2.0

   Property Name:
   com.netegrity.assertiongenerator.saml2.MaxUserAttributeLength

   Property Type: Positive Integer value

   Default Value: 1024

   Description: Indicates the maximum attribute length for SAML2.0 assertion attributes

4. Restart the Policy Server after any change to these parameters.

Use a Script to Create A New Response Attribute

The Advanced section of the Add Attribute page contains the Script field. This field displays the script that CA SiteMinder® generates based on your entries in the Attribute Setup section. You can copy the contents of this field and paste them into the Script field for another response attribute.

Note: If you copy the contents of the Script field to another attribute, select the appropriate option button in the Attribute Kind group.

Customize a SAML Assertion Response (optional)

You can modify the assertion content using an assertion generator plug-in. The plug-in enables you to customize the content of an assertion using the business agreements between you and your partners and vendors. One plug-in is allowed for each partner.

The steps to configure an assertion generator plug-in are:

1. Install the CA SiteMinder® SDK, if you have not done so already.
2. Implement the AssertionGeneratorPlugin.java interface, which is part of the SDK.
3. Deploy your assertion generator plug-in implementation class.
4. Enable the assertion generator plug-in parameters in the Administrative UI.

Additional information about the Assertion Generator plug-in can be found as follows:

- Reference information (method signatures, parameters, return values, data types), and also the new constructor for UserContext class, are in the *Javadoc Reference*. Refer to the AssertionGeneratorPlugin interface in the Javadoc.
- Overview and conceptual information for authentication and authorization APIs is in the *CA SiteMinder® Programming Guide for Java*.

Implement the AssertionGeneratorPlugin Interface

The first step in creating a custom assertion generator plug-in is to implement the AssertionGeneratorPlugin interface.

Follow these steps:

1. Provide a public default constructor method that contains no parameters.
2. Provide code so that the implementation is stateless. Many threads must be able to use a single plug-in class.

3. Implement methods in the interface to satisfy your requirements.

   The implementation must include a call to the customizeAssertion methods. You can overwrite the existing implementations. See the following sample classes for examples:

   SAML 1.x/WS-Federation

   AssertionSample.java

   SAML 2.0

   SAML2AssertionSample.java

   The sample classes are located in the directory /sdk/samples/assertiongeneratorplugin.

   The contents of the parameter string that your implementation passes into the customizeAssertion method is the responsibility of the custom object.

Deploy the Assertion Generator Plug-in

After you have coded your implementation class for the AssertionGeneratorPlugin interface, compile it and verify that CA SiteMinder® can find your executable file.

To deploy the assertion generator plug-in

1. Compile the assertion plug-in Java file.

   Compilation requires the following .jar files, which are installed with the Policy Server:

   - *policy_server_home*/bin/jars/SmJavaApi.jar
   - *policy_server_home*/bin/thirdparty/xercesImpl.jar
   - *policy_server_home*/bin/endorsed/xalan.jar

2. In the JVMOptions.txt file, modify the -Djava.class.path value so it includes the classpath for the plug-in. This modification enables the plug-in to be loaded with the modified classpath. Locate the JVMOptions.txt file in the directory *installation_home*\siteminder\config.

   Note: Do not modify the classpath for xercesImpl.jar, xalan.jar, or SMJavaApi.jar.

3. Enable the plug-in.

Enable the Assertion Generator Plug-in

After writing an assertion generator plug-in and compiling it, enable the plug-in by configuring settings in the Administrative UI. The UI parameters let CA SiteMinder® know where to find the plug-in.

Do not configure the plug-in settings until you deploy the plug-in (see page 143).

Follow these steps:

1. Log on to the Administrative UI.
2. Click Federation, Legacy Federation, Affiliates.
3. Select an existing Affiliate entry or create one.
4. Navigate to the General settings.
5. In the Assertion Generator Plug-in section, complete the following fields:

   Java Class Name

   Specifies a Java class name for an existing plug-in.

   The plug-in class can parse and modify the assertion, and then return the result to the Assertion Generator for final processing.

   Only one plug-in is allowed for each Affiliate. For example, com.mycompany.assertiongenerator.AssertionSample

   Parameter

   (Optional) Specifies a string of parameters that is passed to the plug-in specified in the Java Class Name field.

   Note: Instead of enabling the assertion plug-in in the Administrative UI, you can use the Policy Management API (C or Perl) to integrate the plug-in. For more information, see the *CA SiteMinder® Programming Guide for C* or the *CA SiteMinder® Programming Guide for Java*.

6. Restart the Policy Server.

   Restarting the Policy Server picks up the latest version of the assertion plug-in after being recompiled.

Create Links to Consumer Resources (SAML 1.x)

At the producer, create pages that contain links that direct the user to the consumer site. Each link represents an intersite transfer URL. The user has to visit the intersite transfer URL, where a request to the producer-side Web Agent. The user is then redirected to the Consumer site.

For the SAML artifact profile, the syntax for the intersite transfer URL is:

http://*producer_site*/affwebservices/public/intersitetransfer?SMASSERTIONREF=QUERY&NAME=
affiliate_name&TARGET=http://*consumer_site/target_url*?*query_parameter_name*%3D*query_parameter_value*%26*query_parameter_name*%3D*query_parameter_value*&SMCONSUMERURL=
http://*consumer_site*/affwebservices/public/samlcc&AUTHREQUIREMENT=2

For the SAML POST profile, the syntax for the intersite transfer URL is:

http://*producer_site*/affwebservices/public/intersitetransfer?SMASSERTIONREF=QUERY&NAME=
affiliate_name&TARGET=http://*consumer_site/target_url*

The variables in the intersite transfer URLs are as follows:

producer_site

Specifies the website where the user is authenticated.

affiliate_name

Indicates the name of an affiliate configured in an affiliate domain.

consumer_site

Indicates the site that the user wants to visit from the producer site.

target_url

Target page at the consumer site.

The intersite transfer URLs that the user selects must contain the query parameters listed in the table that follows.

Note: Query parameters for the SAML artifact profile must use HTTP-encoding.

Query Parameter	Meaning
SMASSERTIONREF (required)	For internal use. The value is always QUERY. Do not change this value.
NAME (required)	Name of an affiliate configured in an affiliate domain.
TARGET (required)	The target URL at the consumer site.

Query Parameter	Meaning
SMCONSUMERURL (required only for the artifact profile)	The URL at the consumer site processes the assertion and authenticates the user. For SAML 1.x artifact binding, if a value is specified for the Assertion Consumer URL, it takes precedence over the value of this query parameter.
AUTHREQUIREMENT=2 (required only for the artifact profile)	For internal use. The value is always 2. Do not change this value.

Note: The SAML POST profile does not use SMCONSUMERURL and AUTHREQUIREMENT parameters. However, if you include one of these parameters in the intersite transfer URL you must also include the other.

Example: Intersite transfer URL for the artifact profile:

http://www.smartway.com/affwebservices/public/intersitetransfer?SMASSERTIONREF=QUERY&NAME
=ahealthco&TARGET=http://www.ahealthco.com:85/smartway/index.jsp&SMCONSUMERURL=
http://www.ahealthco.com:85/affwebservices/public/samlcc&AUTHREQUIREMENT=2

Example: Intersite transfer URL for the POST profile:

http://www.smartway.com/affwebservices/public/intersitetransfer?SMASSERTIONREF
=QUERY&NAME=ahealthco&TARGET=http://www.ahealthco.com/index.html

Choosing Whether to Protect the Intersite Transfer URL

The web pages with the intersite transfer URL links can be part of a CA SiteMinder®-protected realm that is configured for persistent sessions. When a user selects one of the links on a protected page, CA SiteMinder® presents the user with an authentication challenge. After the user logs in, a persistent session can be established, which is required to store a SAML assertion.

If these pages are unprotected, the producer directs an affiliate user without a CA SiteMinder® session to an authentication URL. This URL prompts the user to log in to receive a CA SiteMinder® session. Define the Authentication URL when you configure an affiliate in the Administrative UI.

Note: To set up persistent sessions, configure the session store. Set up a session store using the Policy Server Management Console.

Chapter 13: Configure as a SAML 1.x Consumer

Prerequisites for a Relying Partner

For CA SiteMinder® to act as the relying partner, complete following tasks:

- Install the Policy Server.
- Install one of the following components:
  - The Web Agent and the Web Agent Option Pack. The Web Agent authenticates users and establishes a session. The Option Pack provides the Federation Web Services application. Be sure to deploy the FWS application on the appropriate system in your network.
  - The SPS federation gateway, which has an embedded Web Agent and has the Federation Web Services application on the embedded Tomcat web server.

  For more information, see the *Web Agent Option Pack Guid*e.
- Private keys and certificates are imported for functions that require verification and encrypting of messages.
- An asserting partner is set up within the federated network.

How To Configure a SAML 1.x Consumer

Configuring CA SiteMinder® as SAML 1.x consumer requires the following tasks:

1. Complete the SAML 1.x authentication scheme prerequisites.
2. Select the authentication scheme type and assign it a name.
3. Specify the namespace for users being authenticated with the SAML 1.x authentication scheme.
4. Select the single sign-on profile that this consumer supports (artifact or POST).
5. Configure a SAML authentication scheme for each Producer that is a federation partner and generates assertions. Bind each scheme to a realm. The realm must contain the target URLs for federated resources. Protect these resources with a CA SiteMinder® policy.

Tips:

- Certain parameter values at the Producer and Consumer must match for the configuration to work. A list of those parameters is available in Configuration Settings that Must Use the Same Values (see page 361).
- Verify that you are using the correct URLs for the Federation Web Services servlets. The URLs are listed in Federation Web Services URLs Used in SiteMinder Configuration (see page 367).

Optional Configuration Tasks for a Consumer

The following tasks are optional for configuring a consumer:

- Customize assertions using the message consumer plug-in.
- Redirect failed authentication attempts.

Navigating Legacy Federation Dialogs

The Administrative UI provides two ways to navigate to the legacy federation configuration dialogs.

You can navigate in one of two ways:

- Following a wizard to configure a new legacy federation object.

  When you create an object, a page displays with a configuration wizard. Follow the steps in the configuration wizard to create the object.

- Selecting tabs to modify an existing legacy federation object.

  When you modify an existing object, a page displays with a series of tabs. Modify the configuration from these tabs. These tabs are the same as the steps in the configuration wizard.

SAML 1.x Authentication Schemes

A consumer is a site that uses a SAML 1.x assertion to authenticate a user.

Note: A site can be a SAML producer and a SAML consumer.

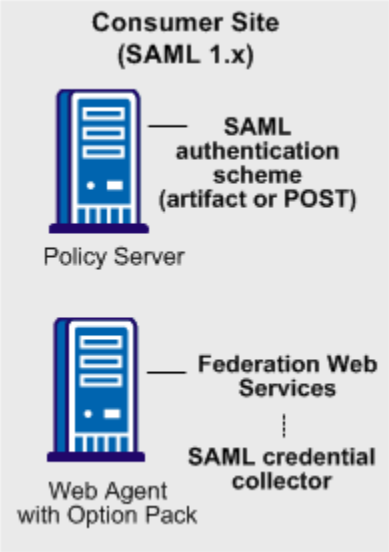
Any CA SiteMinder® site with legacy federation functionality can consume SAML 1.x assertions and can use these assertions to authenticate users. When an assertion is consumed, the site has to be able to compare the information from the assertion against a user directory to complete the authentication process.

CA SiteMinder® provides the following SAML 1.x authentication methods:

- SAML Artifact profile
- SAML POST profile

The SAML-based authentication schemes let a consumer site authenticate a user. Consuming a SAML assertion and establishing a CA SiteMinder® session enables cross-domain single sign-on. After the user is identified, the consumer site can authorize the user for specific resources.

The following illustration shows the major components for authentication at the consumer site.



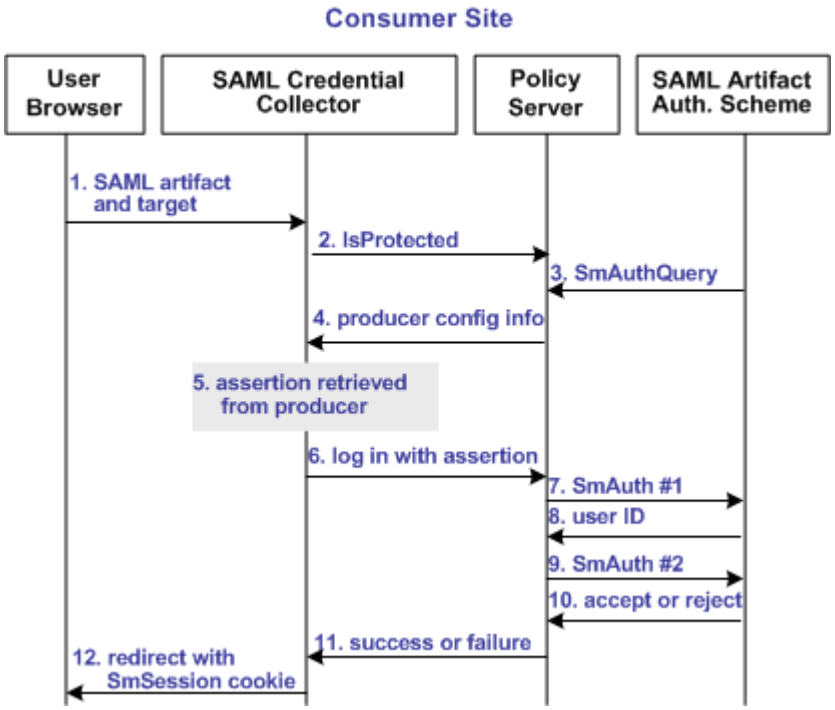
Note: The SPS federation gateway can replace the Web Agent and Web Agent Option Pack to provide the Federation Web Services application functions. For information about installing and configuring the SPS federation gateway, see the *Secure Proxy Server Administration Guide*.

The SAML 1.x authentication scheme is configured at the consumer-side Policy Server. The SAML credential collector is a component of the Federation Web Services application. The credential collector is installed on the consumer-side Web Agent, or on an SPS federation gateway. The credential collector obtains information from the SAML authentication scheme at the Policy Server, then uses that information to access a SAML assertion.

The SAML assertion becomes the credentials that grant access to the Policy Server at the consumer site. The user is authenticated and authorized, and if authorization is successful, the user is redirected to the target resource.

SAML 1.x Artifact Authentication Scheme Overview

The following illustration shows how the SAML 1.x artifact authentication scheme processes requests.



Note: An SPS federation gateway, or the Web Agent and Web Agent Option Pack, provide the Agent and SAML Credential Collector functionality.

Unless otherwise stated, all activity in this process occurs at the Consumer site:

1. A user is redirected to the SAML credential collector with a SAML artifact and a target URL.

   The artifact and target URL are originally generated from the Web Agent at the producer site.

2. The SAML credential collector calls the Policy Server to determine whether the SAML artifact authentication scheme protects the requested resource.

3. The Policy Server passes the necessary data to the SAML artifact authentication scheme, which extracts the producer configuration information.

4. The Policy Server returns the producer configuration information to the SAML credential collector. This information enables the credential collector servlet to call a producer site and retrieve a SAML assertion.

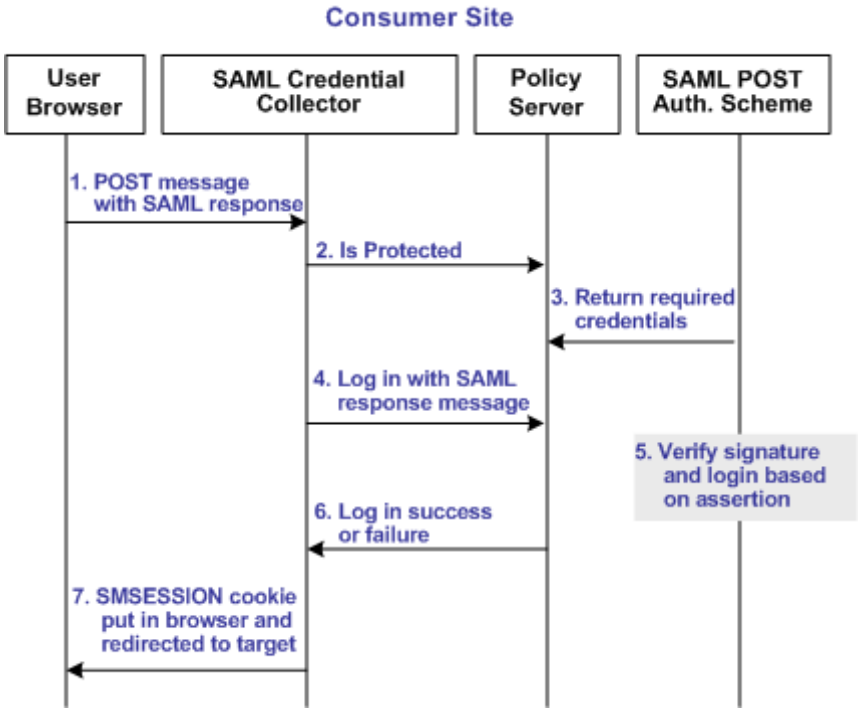
5. The SAML credential collector takes the data from the Policy Server and uses it to retrieve the SAML assertion.

6. Once an assertion is returned, the credential collector uses the assertion as credentials, and logs in to the Policy Server.
7. The Policy Server makes the initial user disambiguation call to the SAML authentication scheme.
8. Using the authentication scheme data and the assertion, the scheme locates the user and returns a unique identifier for the user to the credential collector.
9. The Policy Server makes the second user authentication call to the authentication scheme.

   Note: The CA SiteMinder® Authentication AP dictate the two-step authentication process. For more information, see the *CA SiteMinder® Programming Guide for C* or the *CA SiteMinder® Programming Guide for Java*.
10. The scheme validates the SAML assertion and returns an accept or reject message to the Policy Server.
11. The Policy Server sends the accept or reject message to the credential collector.
12. The SAML credential collector creates a session cookie and places it in the browser, and then redirects the user to the target resource. If the login fails, the credential collector redirects the user to a No Access URL.

SAML 1.x POST Profile Authentication Scheme Overview

The following illustration shows how the SAML 1.x POST profile authentication scheme processes requests.



Note: The SPS federation gateway or the Web Agent Option Pack provide the SAML Credential Collector functionality.

Unless otherwise stated, the following process takes place at the consumer site:

1. A browser posts an HTML form to the SAML credential collector URL. This form contains a SAML response message and the address of the target URL, originally generated at the producer.
2. The SAML credential collector contacts the Policy Server to determine whether the target resource is protected.
3. The Policy Server replies that the SAML POST profile authentication scheme protects the target URL. A signed response from the posted form is the expected credential for the login call.
4. The SAML credential collector makes a login call to the Policy Server, passing the digitally signed SAML response as credentials.
5. The SAML POST profile authentication scheme verifies the signature and other fields of the response and the assertion.

6. If the checks succeed and the user is found in the directory, then authentication succeeds. If any of the checks fail, authentication fails.
7. The SAML credential collector creates an SMSESSION cookie. This cookie is put in the browser and the user is redirected to the target resource. If the login fails, the credential collector redirects the user to the configured No Access URL.

SAML 1.x Authentication Scheme Prerequisites

The prerequisites for configuring a SAML authentication scheme are as follows:

- Install a CA SiteMinder® Policy Server at the producer and the consumer.
- Install Federation Web Services at the producer and the consumer.
- Prepare a certificate data store to sign SAML POST responses.

Install the CA SiteMinder® Policy Server

The CA SiteMinder® Policy Server includes legacy federation functionality.

To install the Policy Server, refer to the *Policy Server Installation Guide*.

Install Federation Web Services at the Producer and Consumer

Federation Web Services (FWS) is a web application. FWS provides the SAML credential collector servlet, which consumes assertions and other services for federated network configurations.

To use the FWS application features, install the Web Agent and Web Agent Option Pack or the SPS federation gateway, which has FWS embedded at the producer and consumer sites.

For installation and configuration instructions, refer to the following guides:

- *Web Agent Option Pack Guide*
- *Secure Proxy Server Administration Guide*

Specify a Value for the DefaultAgentName Setting

When you install a Web Agent, define a value for the Web Agent parameter DefaultAgentName for all consumer Web Agents. This value specifies a Web Agent identity.

Include the specified Agent identifying the DefaultAgentName in the Resource Filter of the realm that protects the target resource. Configure the DefaultAgentName parameter in the Agent Configuration Object or the local Agent configuration file. Omitting the DefaultAgentName parameter or using the value specified in the AgentName parameter in the realm resource filter causes SAML 1.x authentication to fail, regardless of the single sign-on profile.

Set Up a Certificate Data Store to Sign and Verify POST Responses

To use the SAML POST profile for passing assertions, the producer must sign the SAML response that contains the assertion. The assertion consumer at the consumer site must verify that signature.

To accomplish these tasks, add a private key/certificate pair to the certificate data store for signing, verification or both. The certificate data store lets you manage and retrieve keys and certificates, which are required to sign and validate SAML responses.

For more information about the certificate data store, see the *Policy Server Configuration Guide*.

Configure SAML 1.x Artifact Authentication

Before you can assign a SAML artifact authentication scheme to a realm, configure the scheme.

Follow these steps:

1. Navigate to Infrastructure, Authentication, Authentication Schemes.
2. Click Create an authentication scheme.
3. Select Create a new object of type Authentication Scheme.

   The Authentication Scheme page opens.
4. Enter a name for the authentication scheme.

5. From the Authentication Scheme Type drop-down list, select SAML Artifact Template.

   The contents of the Authentication Scheme dialog change to support the SAML artifact scheme.

6. Configure the scheme setup.

   Click Help for descriptions of settings.

   Important! The Affiliate Name, Password, and Verify Password fields must match other values in your federation network. For details, go to Configuration Settings that Must Use the Same Values (see page 361).

7. (Optional) Specify the target resource in the Default Target URL field. This field is in the Additional Configuration section of the page. The target is the protected federated resource at the consumer.

   The consumer does not have to use the default target. The link that initiates single sign-on contains a query parameter that specifies the target.

   Alternatively, specify the target resource using the value of the TARGET query parameter in the authentication response URL. To enable this option, select the checkbox Query Parameter TARGET Overrides Default Target URL.

8. (Optional) Configure features, such as the Message Consumer API and redirect URLs for authentication errors in the Additional Configuration section.

9. Click OK to save the scheme.

The SAML 1.x Artifact authentication scheme is now configured.

More information:

Back Channel Configuration for HTTP-Artifact SSO (see page 156)

Back Channel Configuration for HTTP-Artifact SSO

For the SAML artifact profile, the asserting party sends the assertion to the consumer over a back channel. Protect the back channel with an authentication scheme. You can use a basic or client certificate authentication scheme to secure the back channel.

- Basic authentication

  If you use basic authentication and CA SiteMinder® is at both partners, the Affiliate Name at each site is the name of the consumer. If the asserting party is not CA SiteMinder®, the asserting party administrator must provide you with the name they are using to identify your site. Specify the supplied name as the Affiliate Name in your authentication scheme configuration.

- Client certificate authentication

  If you use client certificate authentication for the back channel, the affiliate name in the Administrative UI must be the alias of the client certificate. Additionally, the CN of the certificate subject must also match the affiliate name. Matching the affiliate name, alias and CN is required.

The Policy Server supports client certificate authentication over the backchannel using non-FIPS 140 encrypted certificates, even when the Policy Server is operating in FIPS-only mode. However, for a strictly FIPS-only installation, use certificates only encrypted with FIPS 140-compatible algorithms.

The client certificate is stored in the certificate data store.

Configure SAML 1.x POST Profile Authentication

To configure the SAML POST profile authentication scheme

1. Navigate to Infrastructure, Authentication, Authentication Schemes.
2. Click Create an authentication scheme.
3. Select Create a new object of type Authentication Scheme.

   The Authentication Scheme page opens.
4. Enter a name for the authentication scheme.
5. From the Authentication Scheme Type drop-down list, select SAML POST Template.

   The contents of the Authentication Scheme dialog change to support the SAML POST scheme.

6. Configure the scheme setup.

   Click Help for the field descriptions.

   Important! The Affiliate Name, Password, and Verify Password fields must match other values in your federation network. For details, go to Configuration Settings that Must Use the Same Values (see page 361).

7. (Optional) Specify the target resource in the Default Target URL field. This field is in the Additional Configuration section of the page. The target is the protected federated resource at the consumer.

   The consumer does not have to use the default target. The link that initiates single sign-on contains a query parameter that specifies the target.

   Alternatively, specify the target resource using the value of the TARGET query parameter in the authentication response URL. To enable this option, select the checkbox Query Parameter TARGET Overrides Default Target URL.

8. (Optional) Configure features, such as the Message Consumer API and redirect URLs for authentication errors in the Additional Configuration section.

9. Click OK to save the scheme.

The SAML 1.x POST authentication scheme is now configured.

Customize Assertion Processing with the Message Consumer Plug-in

The message consumer plug-in is a Java program that implements the Message Consumer Plug–in. The plug-in lets you implement your own business logic for processing assertions, such as rejecting an assertion and returning a status code. This additional processing works together with the standard processing of an assertion.

Note: For more information about status codes for authentication and disambiguation, see the *CA SiteMinder® Programming Guide for Java*.

During authentication, CA SiteMinder® first tries to process the assertion by mapping a user to its local user store. If CA SiteMinder® cannot find the user, it calls the postDisambiguateUser method of the message consumer plug-in.

If the plug-in successfully finds the user, CA SiteMinder® proceeds to the second phase of authentication. If the plug-in cannot map the user to a local user store, the plug-in returns a UserNotFound error. The plug-in can optionally use the redirect URL feature. Without the consumer plug-in, the redirect URLs are based on the error that the SAML authentication scheme generates.

During the second phase of authentication, CA SiteMinder® calls the postAuthenticateUser method of the message consumer plug-in, if the plug-in is configured. If the method succeeds, CA SiteMinder® redirects the user to the requested resource. If the method fails, you can configure the plug-in to send the user to a failure page. The failure page can be one of the redirect URLs that you can specify with the authentication scheme configuration.

Additional information about the message consumer plug-in can be found as follows:

- Reference information (method signatures, parameters, return values, data types), and the constructor for UserContext class, are in the *Java Developer Reference*. Refer to the MessageConsumerPlugin interface.
- Overview and conceptual information about authentication and authorization APIs, see the *CA SiteMinder® Programming Guide for Java*.

To configure the plugin

1. Install the CA SiteMinder® SDK, if you have not done so already.
2. Implement the MessageconsumerPlugin.java interface, which is part of the CA SiteMinder® SDK.
3. Deploy your message consumer plug-in implementation class.
4. Enable the message consumer plug-in in the Administrative UI.

Implement the MessageConsumerPlugin Interface

Create a custom message consumer plug-in by implementing the MessageConsumerPlugin.java interface. The minimum requirements for the implementation class are listed in the following procedure.

Follow these steps:

1. Provide a public default constructor method that contains no parameters.
2. Provide code so that the implementation is stateless. Many threads must be able to use a single plug-in class.

3. Implement methods in the interface as your requirements demand.

   The MessageConsumerPlugin includes the following four methods:

   init()

   Performs any initialization procedures that the plug-in requires. CA SiteMinder® calls this method once for each plug-in instance, when the plug-in is loaded.

   release()

   Performs any rundown procedures that the plug-in requires. CA SiteMinder® calls this method once for each plug-in instance, when CA SiteMinder® is shutting down.

   postDisambiguateUser()

   Provides processing to disambiguate a user when the authentication scheme is unable to do so. Alternatively, this method can add data for new federation users to a user store. This method receives the decrypted assertion. The decrypted assertion is added to the properties map passed to plug-in under the key "_DecryptedAssertion".

   postAuthenticateUser()

   Provides additional code to determine the final outcome of assertion processing, regardless of whether the Policy Server processing is a success or failure.

CA SiteMinder® provides the following samples of the Message Consumer plug-in class:

MessageConsumerPluginSample.java in *installation_home*\sdk\samples\messageconsumerplugin

MessageConsumerSAML20.java in *installation_home*\sdk\samples\authextensionsaml20

Deploy a Message Consumer Plug-in

After you have coded your implementation class for the MessageConsumerPlugin interface, compile it and verify that CA SiteMinder® can find your executable file.

To deploy the Message Consumer Plugin:

1. Compile the MessageConsumerPlugin Java file. The file requires the following dependent libraries, which are installed with the Policy Server:

   installation_home\siteminder\bin\jars\SmJavaApi.jar

   An identical copy of SmJavaApi.jar is installed with CA SiteMinder® SDK. The file is in the directory *installation_home*\sdk\java\SmJavaApi.jar.

   You can use either of them at development time.

2. When a plug-in class is available, in a folder or a jar file, modify the -Djava.class.path value in the JVMOptions.txt file. This step enables the plug-in class to load with the modified classpath. Locate the JVMOptions.txt file in the directory *installation_home*\siteminder\config.

   Note: Do not modify the classpath for the existing xerces.jar, xalan.jar, or SmJavaApi.jar.

3. Restart the Policy Server to pick up the latest version of MessageConsumerPlugin. This step is necessary each time the plug-in Java file is recompiled.

4. Enable the plug-in.

Enable the Message Consumer Plug-in for SAML 1.x

After writing a message consumer plug-in and compiling it, enable the plug-in by configuring settings in the Administrative UI. The UI settings tell CA SiteMinder® where to find the plug-in.

Do not configure the plug-in settings until you deploy the plug-in (see page 159).

To enable the message consumer plug-in

1. Log on to the Administrative UI

2. Navigate to the Authentication Scheme dialog for the appropriate SAML 1.x scheme. In the Additional Configuration section, complete the following fields:

   Full Java Class Name

   Specify the Java class name for the plug-in, For example, a sample class included with the CA SiteMinder® SDK is:

   com.ca.messageconsumerplugin.MessageConsumerPluginSample

   Parameter

   Specify a string of parameters that are passed to the plug-in specified in the Full Java Class Name field.

   As an alternative to configuring the plug-in in the Administrative UI, use the Policy Management API (C or Perl) to set the IdpPluginClass and IdpPluginParameters.

3. Restart the Policy Server.

Redirect Users After Failed SAML 1.x Authentication Atempts

If a consumer cannot authenticate a user during a single sign-on transaction, the consumer can redirect that user to a customized URL for further processing.

You can configure several optional redirect URLs for failed authentication. These redirect URLs allow you more control over where a user is redirected. For example, if a user cannot be located in a user store, you can fill in a User Not Found redirect URL.

The Status Redirect URLs and Modes are in the Additional Configuration section of the authentication dialog. The redirect URLs are for specific status conditions:

- User is not found
- Single sign-on message is invalid
- User credentials are not accepted

If any of the conditions occur, redirect URLs can send the user to an application or a customized error page for further action.

Note: Configuring redirect URLs is not required.

If you do not configure redirect URLs, standard CA SiteMinder® processing takes place. How a failed authentication is handled depends on the configuration of the authentication scheme.

To configure status redirect URLs

1. Navigate to the page for a SAML Artifact or SAML POST authentication scheme.
2. In the Status Redirect URLs and Modes section, fill in a URL for one or more of the fields.

   Click Help for descriptions of settings.

   Federation Web Services handles the errors by mapping the authentication reason into one of the configured redirect URLs. The user can be redirected to that URL to report the error.
3. Select one of the following modes:
   - 302 No Data
   - HTTP POST
4. Click OK to save your changes.

Note: These redirect URLs can be used with the CA SiteMinder® Message Consumer Plug-in for further assertion processing. If authentication fails, the plug-in can send the user to one of the redirect URLs you specify.

More information:

Configure SAML 1.x Artifact Authentication (see page 154)
Configure SAML 1.x POST Profile Authentication (see page 156)

Supply SAML Attributes as HTTP Headers

An assertion response can include attributes in the assertion. These attributes can be supplied as HTTP header variables so a client application can use them for finer grained access control.

The benefits of including attributes in HTTP headers are as follows:

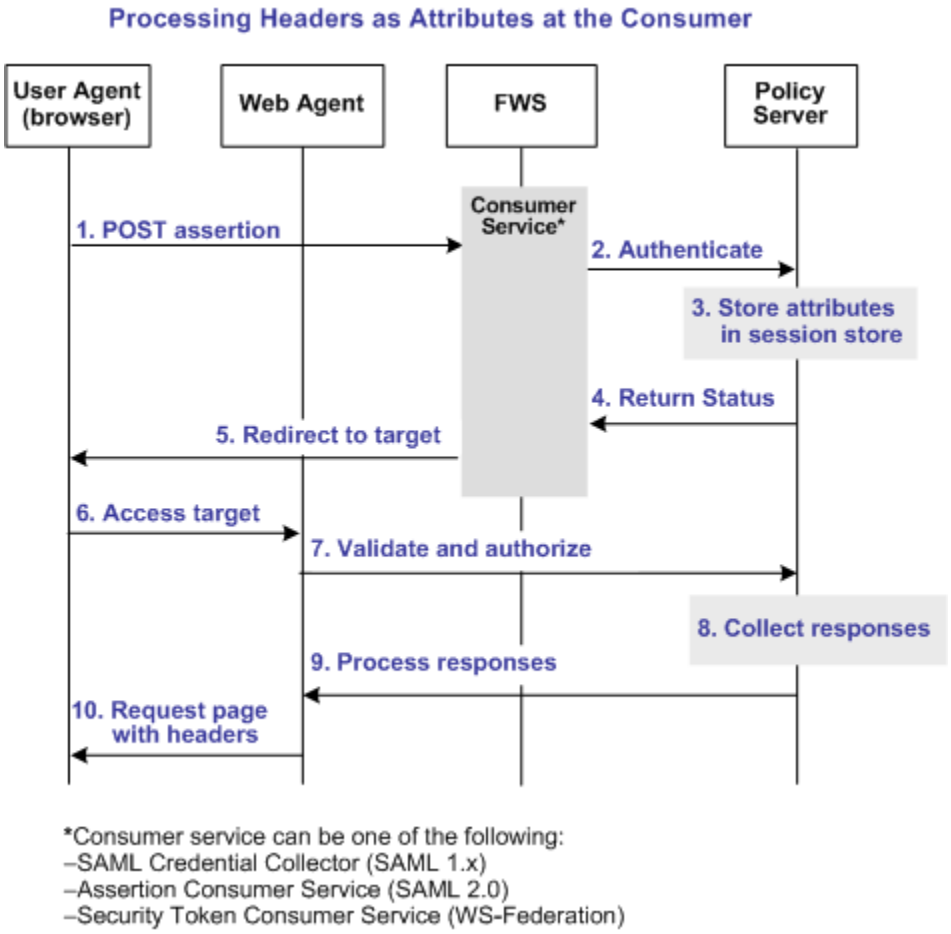
- HTTP headers are not persistent. They are present only within the request or response that contains them.
- HTTP headers, as supplied by the CA SiteMinder® Web Agent, are not visible in the browser, which reduces security concerns.

Note: The HTTP headers have size restrictions that the attributes cannot exceed. CA SiteMinder® can send an attribute in a header up to the web server size limit for a header. Only one assertion attribute per header is allowed. See the documentation for your web server to determine the header size limit.

Use Case for SAML Attributes As HTTP Headers

During authentication, a series of SAML attributes are extracted from an assertion and supplied as HTTP headers. During the authorization process, these headers are returned to the customer application.

The following flow diagram shows the sequence of events at runtime:



Note: The SPS federation gateway can replace the Web Agent and Web Agent Option Pack to provide the Federation Web Services application functions. In the flow diagram, the Web Agent block would be the embedded Web Agent in the SPS federation gateway. For information about installing and configuring the SPS federation gateway, see the *Secure Proxy Server Administration Guide*.

To process the attributes as HTTP headers, the sequence of events is as follows:

1. After the assertion is generated at the asserting party, it sends the assertion to the appropriate consumer service at the relying party. The delivery mechanism (POST or Artifact or WS-Fed) is irrelevant.

   Note: The consumer service can be the SAML credential collector (SAML 1.x), the Assertion Consumer Service (SAML 2.0), or Security Token Consumer Service (WS-Federation).

2. The consumer service calls its local Policy Server to use the configured authentication scheme to authenticate the user with the assertion.

3. If the authentication scheme redirect mode parameter is set to PersistAttributes, the Policy Server caches the attributes in the session store as session variables.
4. The result of the authentication is returned to the consumer service.
5. The consumer service redirects the browser to the protected target resource.
6. The browser tries to access the target resource.
7. The Web Agent calls the Policy Server to validate the user session and to verify that the user is authorized to access the target resource.
8. The Policy Server retrieves the attributes by a configured response.
9. The Policy Server processes the responses and sends the attributes to the Web Agent.
10. The Web Agent sets the HTTP headers as necessary.

Configuration Overview to Supply Attributes as HTTP Headers

Several configuration steps are required to retrieve the SAML attributes cached in the session store and provide them as HTTP headers.

Follow these steps:

1. Select PersistAttributes as the redirect mode for the SAML authentication scheme, which enables the SAML Attributes to be returned as HTTP headers.
2. Configure an authorization rule for the realm that contains the target resource.
3. Set PersistentRealm in the realm protecting the target resource.
4. Configure a response that uses the active response type for each SAML attribute to be supplied as a header.
5. Create a policy that binds the authorization rule and active response to implement the user of attributes as HTTP headers.

Set the Redirect Mode to Store SAML Attributes

After the relying party authenticates the user with the SAML assertion, the SAML attributes are written to the session store. The browser is then redirected to the target resource.

To redirect the browser with the attribute data

1. Log in to the Administrative UI.
2. Navigate to the configuration page of the SAML authentication scheme.

3. Set the Redirect Mode parameter to Persist Attributes. Locate the Redirect Mode field as follows:

   SAML 1.x

   The Redirect Mode is in the Scheme Setup section of the main configuration page.

   SAML 2.0

   Click SAML 2.0 Configuration, SSO. The Redirect Mode is in the SSO section of the page.

   WS-Federation

   Click WS-Federation Configuration, SAML Profiles. The Redirect Mode is in the SSO section of the page.

4. Click Submit to save your changes.

The redirect mode is now set to pass on the attribute data.

Create an Authorization Rule to Validate Users

For the realm containing the protected target resource, create a rule to retrieve the SAML attributes from the session store.

The rule is based on an authorization event (onAccessAccept). The user is already authenticated by the FWS application. The Web Agent cannot reauthenticate the user and then pass on the HTTP headers. The retrieval of the attributes occurs during the authorization stage.

To create an OnAccessAccept Rule for the realm

1. Log on to the Administrative UI.
2. Navigate to Policies, Domain, Realms.
3. Select the realm with the target resource.
4. Click Create in the Rules section.

   The Create Rule page appears.

5. Enter a name and optionally, a description.
6. Enter an asterisk (*) in the Resource field.
7. Select Authorization events and OnAccessAccept in the Action section.
8. Select Enabled in the Allow/Deny and Enable/Disable section.
9. Click OK to save the rule.

The authorization rule is now defined for the realm with the protected resource.

Configure a Response to Send Attributes as HTTP Headers

Configure a response that sends the SAML attributes as HTTP headers to the Web Agent. The Web Agent processes the response and makes the header variables available to the client application.

Follow these steps:

1. Log on to the Administrative UI.
2. Navigate to Policies, Domain, Domains.
3. Select the domain for the target resource and click Modify.
4. Select the Responses tab.
5. Click Create.

   The Response dialog opens.
6. Enter a name.
7. Confirm that the Agent type is a CA SiteMinder® Web Agent.
8. Click Create Response Attribute.

   The Response Attribute dialog opens.
9. Select WebAgent-HTTP-Header-Variable in the Attribute field.
10. Select Active Response for Attribute Kind.
11. Complete the fields as follows:

    Variable Name

    Specify the name that you want for the header variable. You assign this name.

    Library Name

    smfedattrresponse

    This value must be the entry for this field.

    Function Name

    getAttributeValue

    This value must be the entry for this field.

    Parameters

    Specify the name of the attribute as it appears in the assertion.

    An agreement between you and your federated partner determines the attributes that are in the assertion.
12. Click OK to save the attribute.

13. Repeat the procedure for each attribute that is to become an HTTP header variable. You can configure many attributes for a single response.

    You return to the Response tab. The attributes that you create are listed in the Attributes List section.

14. Click OK to save the response.

    You return to the Response tab.

15. Click Submit to save the domain.

The response sends the attributes on to the Web Agent to become HTTP headers.

Create a Policy to Implement Attributes as HTTP Headers

To implement the use of SAML attributes as HTTP headers, group together the authorization event rule and active response in a policy.

Follow these steps:

1. Log on to the Administrative UI.
2. Navigate to Policies, Domain, Domains.
3. Select the domain that contains the target resource and click Modify.
4. Select the Policy tab and click Create in the Policy section.

   The Create Policy dialog opens.

5. Enter a descriptive name in the Name field.
6. Select the users who are to have access to the protected resource in the Users tab.
7. Add the authorization rule that you created previously on the Rules tab.
8. Select the authorization rule and click Add Response.

   The Available Responses dialog opens.

9. Select the active response that you created previously and click OK.

   You return to the Rules tab. The response appears with the authentication rule.

10. Click Submit to save the policy.

The policy that enables SAML attributes to be used as HTTP headers is complete.

Enable Client Certificate Authentication for the Back Channel (optional)

If you are using HTTP-Artifact single sign-on, you can select client certificate authentication to protect the Assertion Retrieval Service at the producer. This service retrieves the assertion and sends it to the consumer.

Note: Client certificate authentication is optional; you can also use Basic authentication.

The SAML credential collector invokes the SAML artifact authentication scheme. The SAML credential collector collects information from the scheme to retrieve the SAML assertion from the Producer. You are required to specify the authentication method for the realm that contains the Assertion Retrieval Service. The SAML credential collector determines what type of credentials to provide to retrieve the assertion.

If the Assertion Retrieval Service is protected with a client certificate authentication scheme, complete these configuration tasks:

1. Add a client certificate to the certificate data store (see page 168).
2. Select the client certificate option for back channel authentication (see page 169). The certificate is the required credential.

Note: The administrator at the asserting-side Policy Server must have configured a policy to protect the Assertion Retrieval Service. The realm for this policy must use an X.509 client certificate authentication scheme.

Add a Client Certificate to the Certificate Data Store

You must have a private key/certificate pair from a Certificate Authority. Add a private key/certificate pair to the certificate data store using the Administrative UI. Skip this step if the key/certificate pair is already in the data store. For instructions, see the *Policy Server Configuration Guide.*

When you import the key/certificate pair, the alias you assign must be the same value as the Affiliate Name field in the authentication scheme settings. Additionally, the CN attribute of the Subject in the certificate must also match the Affiliate Name field. For example, the Affiliate Name is CompanyA. Therefore, the alias must be Company A, and the CN value for the Subject must read CN=CompanyA, OU=Development, O=CA, L=Islandia, ST=NY, C=US.

Important! The Affiliate Name field in the authentication scheme must match the name that is assigned to the affiliate object at the Producer. If CA SiteMinder® is the Producer, the Affiliate Name in the authentication scheme must match the Name field in the General settings of the affiliate object.

Select the Client Cert Option for Back Channel Authentication

For the Consumer to present a certificate as credentials when trying to access the Assertion Retrieval Service at the Producer, select the client certificate option.

To select the client certificate option

1. Go to the Scheme Setup section of the SAML Artifact Authentication scheme dialog.
2. Select Client Cert for the Authentication field.

How To Protect a Resource with a SAML 1.x Authentication Scheme

Protect target federation resources by configuring a CA SiteMinder® policy that uses the SAML 1.x authentication scheme.

Follow these steps:

1. Create a realm that uses the SAML authentication scheme. The realm is the collection of target resources.

   You can create a realm in the following ways:

   - Create a unique realm (see page 170) for each authentication scheme already configured.
   - Configure a single target realm (see page 170) that uses a custom authentication scheme to dispatch requests to the corresponding SAML authentication schemes. Configuring one realm with a single target for all producers simplifies configuration of realms for SAML authentication.
2. Configure an associated rule and optionally, a response.
3. Group the realm, rule, and response into a policy that protects the target resource.

Important! Each target URL in the realm is also identified in an intersite transfer URL. The intersite transfer URL redirects a user from the producer to the consumer. You specify this URL in the URL TARGET variable. At the producer site, an administrator includes this URL in a link that redirects the user to the consumer.

Configure a Unique Realm for Each Authentication Scheme

The procedure for configuring a unique realm for each SAML or WS-Federation authentication scheme follows the standard instructions for creating realms.

Follow these steps:

1. Navigate to Policy, Domain, Domains.

   The page to create domains displays.

2. Click Create Domain.

3. Enter a domain name.

4. Add the user directory to the domain. This directory is the one that contains the users requesting access to federated resources.

5. Select the Realm tab and create a realm.

   - In the Agent field, select the Web Agent protecting the web server where the target resources reside.
   - Select the appropriate authentication scheme in the Authentication Scheme field.

6. Create a rule for the realm.

   As part of the rule, select an action (Get, Post, or Put) that allows you to control processing when users authenticate.

7. Select the Policies tab and configure a policy that protects the target federation resource. Associate the realm that you previously created with this policy.

A policy with a unique realm now protects the federated resources.

Configure a Single Target Realm for All Authentication Schemes

To simplify configuration of realms for authentication schemes, create a single target realm for multiple sites generating assertions.

To do this task, set up the following components:

- A single custom authentication scheme

  This custom scheme forwards requests to the corresponding SAML or WS-Federation authentication schemes that you already configured for each asserting party.

- A single realm with one target URL

Create Authentication Schemes for the Single Target Realm

To define a custom authentication scheme for a single target realm, you must:

- Configure the authentication schemes.
- Define a parameter in the custom scheme that tells the Policy Server which authentication schemes to apply to resource requests.

First, verify that there are configured SAML or WS-Federation authentication schemes. If not, configure these schemes that the custom scheme can reference.

To create the authentication scheme

1. Navigate to Infrastructure, Authentication, Authentication Schemes.

   The Create Authentication Scheme page appears.

2. Create one or more authentication schemes according to the procedures for the protocol you are using.
3. Click OK to exit.

More information:

SAML 1.x Authentication Schemes (see page 148)
WS-Federation Authentication Scheme Overview (see page 297)
How to Configure a SAML 2.0 Authentication Scheme (see page 235)

Create the Custom Authentication Scheme

A single target realm relies on a specific custom authentication scheme to work properly.

To configure a custom authentication scheme for a single target realm

1. Navigate to Infrastructure, Authentication, Authentication Schemes.

   The Create Authentication Scheme page appears.

2. Complete the fields as follows:

   Name

   Enter a descriptive name for the custom authentication scheme, such as SAML Custom Auth Scheme.

3. In the Scheme Common Setup section, complete the following fields:

   Authentication Scheme Type

   Custom Template

   Protection Level

   Accept the default of set a new level.

4. In the Scheme Setup section, complete the following fields:

   Library

   smauthsinglefed

   Secret

   Leave this field blank.

   Confirm Secret

   Leave this field blank.

   Parameter

   Specify one of the following parameters:

   - SCHEMESET=LIST; <*saml-scheme1*>;<*saml_scheme2*>

     Specifies the list of SAML authentication scheme names to use. If you configured an artifact scheme named artifact_producer1 and POST profile scheme named samlpost_producer2, you enter these schemes. For example:

     SCHEMESET=LIST;artifact_producer1;samlpost_producer2

   - SCHEMESET=SAML_ALL;

     Specifies all the configured schemes. The custom authentication scheme enumerates all the SAML authentication schemes and finds the one with the correct Provider Source ID for the request.

   - SCHEMESET=SAML_POST;

     Specifies all the SAML POST Profile schemes that you have configured. The custom authentication scheme enumerates the POST Profile schemes and finds the one with the correct Provider Source ID for the request.

   - SCHEMESET=SAML_ART;

     Specifies all the SAML artifact schemes that you have configured. The custom authentication scheme enumerates the artifact schemes and finds the one with the correct Provider Source ID for the request.

   - SCHEMESET=WSFED_PASSIVE;

     Specifies all the WS-Federation authentication schemes to find the one with the correct Account Partner ID.

Enable this scheme for CA SiteMinder® Administrators

Leave unchecked.

5. Click Submit.

The custom authentication scheme is complete.

Configure the Single Target Realm

After you configure the authentication schemes and associate them with a custom scheme, configure a single target realm for federation resources.

Follow these steps:

1. Navigate to Policies, Domain, Domains.
2. Modify the policy domain for the single target realm.
3. Select the Realms tab and click Create.

   The Create Realm dialog opens.
4. Enter the following values to create the single target realm:

   Name

   Enter a name for this single target realm.
5. Complete the following field in the Resource option:

   Agent

   Select the Web Agent protecting the web server with the target resources.

   Resource Filter

   Specify the location of the target resources. The location is where any user requesting a federated resource gets redirected.

   For example, /FederatedResources.
6. Select the Protected option in the Default Resource Protection section.
7. Select the previously configured custom authentication scheme in the Authentication Scheme field.

   For example, if the custom scheme was named Fed Custom Scheme, you would select this scheme.
8. Click OK.

The single target realm task is complete.

Configure the Rule for the Single Target Realm

After you configure the single target realm, configure a rule to protect the resources.

1. Navigate to the Modify page for the single target Realm.
2. Click Create in the Rules section.

   The Create Rule page appears.
3. Enter values for the fields on the rules page.
4. Click OK.

The single target realm configuration includes the new rule.

Create a Policy Using the Single Target Realm

Create a policy that references the single target realm. Remember that the single target realm uses the custom authentication scheme that directs requests to the appropriate SAML authentication scheme.

Note: This procedure assumes that you have already configured the domain, custom authentication scheme, single target realm and associated rule.

Follow these steps:

1. Navigate to the previously configured domain.
2. Select the Policies tab and click create.

   The Create Policy page opens.
3. Enter a name and a description of the policy in the General section.
4. Add users to the policy from the Users section.
5. Add the rule that you created for the single target realm from the Rules tab.

   The remaining tabs are optional.
6. Click OK.
7. Click Submit.

The policy task is complete. When a request triggers this policy, it relies on the single realm and associated authentication schemes to authenticate the user.

Chapter 14: Configure a SAML 2.0 Identity Provider

Prerequisites for an Asserting Partner(legacy)

To configure an asserting partner, verify the following conditions:

- The Policy Server is installed.
- One of the following options is installed:
  - The Web Agent and the Web Agent Option Pack. The Web Agent authenticates users and establishes a CA SiteMinder® session. The Option Pack provides the Federation Web Services application. Be sure to deploy the FWS application on the appropriate system in your network.
  - The SPS federation gateway has an embedded Web Agent and the Federation Web Services application on the embedded Tomcat web server.

  For more information, see the *Web Agent Option Pack Guid*e.
- Private keys and certificates are imported for functions that require signing and decrypting messages.
- A relying partner is set up within the federated network.

How to Configure an Identity Provider

CA SiteMinder®, as an Identity Provider generates assertions for its business partners, the Service Providers. To establish a federated partnership, the Identity Provider needs information about each partner. Create a Service Provider object for each partner and define how the two entities communicate to pass assertions and to satisfy profiles, such as single sign-on.

To configure an Identity Provider

1. Create a Service Provider object.
2. Add the Service Provider to an affiliate domain.
3. Specify the general identifying information for the Service Provider.
4. Select users from a user store. The Identity Provider generates assertions for these users.

5. Specify the Name ID.
6. Configure a single sign-on (SSO) profile.

   You can save a Service Provider entity without configuring a complete SSO profile. However, you cannot pass an assertion to the Service Provider without completing the SSO configuration.
7. Configure signing and encryption for requests and responses.
8. Complete optional configuration tasks.

Tips:

- Certain parameter values at the Identity Provider and Service Provider must match for the configuration to work. A list of those parameters can be found in Configuration Settings that Must Use the Same Values (see page 361).
- Use the correct URLs for the Federation Web Services servlets. A list of URLs can be found in Federation Web Services URLs Used in SiteMinder Configuration (see page 367)

Optional Configuration Tasks for Identifying a Service Provider

The following optional tasks are for identifying a Service Provider:

- Configure IP address restrictions (see page 181) to limit the addresses that are used to access Service Providers.
- Configure time restrictions (see page 180) for Service Provider operations.
- Enable enhanced client or proxy profile (see page 190).
- Configure attributes for inclusion in assertions.
- Configure single logout (SLO) (see page 222).
- Configure the Identity Provider Discovery profile (see page 225).
- Encrypt the Name ID (see page 228) in the assertion and/or the entire assertion.
- Sign the assertion (see page 197) and/or the entire assertion response.
- Sign the artifact resolve message (see page 197) and/or the artifact response.
- Customize a SAML assertion respons (see page 142)e using the Assertion Generator plug-in.

Navigating Legacy Federation Dialogs

The Administrative UI provides two ways to navigate to the legacy federation configuration dialogs.

You can navigate in one of two ways:

- Following a wizard to configure a new legacy federation object.

  When you create an object, a page displays with a configuration wizard. Follow the steps in the configuration wizard to create the object.

- Selecting tabs to modify an existing legacy federation object.

  When you modify an existing object, a page displays with a series of tabs. Modify the configuration from these tabs. These tabs are the same as the steps in the configuration wizard.

Add a SAML 2.0 Service Provider to an Affiliate Domain

To identify a Service Provider as an available consumer of CA SiteMinder®-generated assertions, add the Service Provider to an affiliate domain at the Identity Provider. You then define the configuration of the Service Provider so that the Identity Provider can issue assertions for it.

Follow these steps:

1. Log in to the Administrative UI.
2. Click Federation, Legacy Federation, SAML Service Providers.
3. Click Create SAML Service Provider.
4. Select an Affiliate Domain then click Next.

Configure the general settings.

Configure General Information for the Service Provider Object

Select the General page to name the Service Provider and provide details such as the SP ID and IDP ID. In addition, you can configure IP address and time restrictions for accessing a Service Provider.

To configure the general settings

1. Navigate to the General settings.

2. Fill in values for the fields, noting the required fields.

   Click Help for the field descriptions, but note the following fields:

   Authentication URL

   This URL points to the redirect.jsp file. Protect the redirect.jsp file with a CA SiteMinder® policy. The policy triggers an authentication challenge to users who request a protected Service Provider resource but do not have a CA SiteMinder® session.

   Skew Time

   Specifies the difference, in seconds, between the system clock at the Identity Provider and the system clock at the Service Provider. Skew Time is used for single sign-on and single logout.

   For single sign-on, the value of the Skew Time and the single sign-on validity duration (Validity Duration field on the SSO tab) determine how long an assertion is valid. Review how the assertion validity is calculated (see page 128) to understand more about the skew time.

   For single logout, the values of the Skew Time and the SLO validity duration (Validity Duration field on the SLO tab) determine the total time that the single logout request is valid. Review how the single logout request validity (see page 223) is calculated to understand more about the skew time.

More Information:

Configure IP Address Restrictions for Service Providers (optional) (see page 181)
Configure Time Restrictions for Service Provider Availability (optional) (see page 180)

Authenticate Users with No CA SiteMinder® Session

When you add a Service Provider to an affiliate domain, one of the parameters you are required to set is the Authentication URL parameter.

The Authentication URL points to the redirect.jsp file. This file is installed at the Identity Provider site where you install the Web Agent Option Pack or the SPS federation gateway. Protect the redirect.jsp file with a CA SiteMinder® policy. The policy triggers an authentication challenge to users who request a protected Service Provider resource but do not have a CA SiteMinder® session.

A CA SiteMinder® session is required for the following bindings:

- For users requesting a protected Service Provider resource

  If you configure single sign-on using an HTTP artifact binding, set up a persistent session to store SAML assertions in the session store.

- For single sign-on using an HTTP POST binding

  A user must have a session; however, the session does not have to be persistent. Assertions are delivered directly to the Service Provider through the browser. The assertions do not have to be stored in the session store.

- For single logout

  If you enable single logout, a persistent session is required. When a user first requests a Service Provider resource, the session is stored in the session store. The session information is necessary when a single logout is later executed.

After a user is authenticated and successfully accesses the redirect.jsp file, a session is established. The redirect.jsp file redirects the user back to the Identity Provider Web Agent or the SPS federation gateway. CA SiteMinder® then processes the request.

The procedure for protecting the Authentication URL is the same regardless of the following deployments:

- Web Agent Option Pack installed on the same system as the Web Agent
- Application server with a Web Agent installed on a web server proxy
- Application server with an Application Server Agent
- SPS federation gateway that is installed at the Identity Provider

Configure a Policy to Protect the Authentication URL

To protect the Authentication URL

1. Log in to the Administrative UI.
2. Create Web Agents to bind to the realms that you define for the asserting party web server. Assign unique agent names for the web server and the FWS application or use the same agent name for both.
3. Create a policy domain for the users who are challenged when they try to access a consumer resource.
4. Select the users that must have access to the resources that are part of the policy domain.

5. Define a realm for the policy domain with the following values:

   Agent

   Agent for the asserting party web server

   Resource Filter

   Web Agents r6.x QMR 6, r12.0 SP2, r12.0 SP3 and SPS federation gateway enter:

   /siteminderagent/redirectjsp/

   The resource filter /siteminderagent/redirectjsp/ is an alias that the FWS application sets up automatically. The alias references include:

   - Web Agent:

     web_agent_home/affwebservices/redirectjsp

   - SPS federation gateway:

     sps_home/secure-proxy/Tomcat/webapps/affwebservices/redirectjsp

   Persistent Session

   For the SAML artifact profile only, select the Persistent check box in the Session section of the realm dialog. If you do not configure a persistent session, the user cannot access consumer resources.

   For the remaining settings, accept the defaults or modify as needed.

6. Click OK to save the realm.
7. Create a rule for the realm. In the Resource field, accept the default value, the asterisk (*), to protect all resources for the realm.
8. Create a policy for the asserting party web server that includes the rule created in the previous step.
9. Complete the task Select Users for Which Assertions are Generated (see page 123).

Configure Time Restrictions for Service Provider Availability (optional)

You can specify time restrictions for when a Service Provider resource is available. When you specify a time restriction, access to the resource is only during the period specified. If a user attempts to access a resource outside of the designated period, the Identity Provider does not generate a SAML assertion.

Note: Time restrictions are based on the system clock of the server on which the Policy Server is installed.

To specify a time restriction

1. Begin at the General settings.

   In the Restrictions section of the page, click Set in the Time section.

   The Time Restriction page displays.

2. Complete the schedule. This schedule grid is identical to the Time Restriction grid for rule objects. For more information, see the *Policy Server Configuration Guide*.

3. Click OK.

The time restriction schedule is set.

Configure IP Address Restrictions for Service Providers (optional)

You can specify an IP address, range addresses, or a subnet mask of the web server where the browser is running to access a Service Provider. If IP addresses are specified for a Service Provider, the Service Provider only accepts user from the appropriate IP addresses.

To specify IP addresses

1. Begin at the General settings.

   In the Restrictions section of the page, click Add in the IP Address area.

   The IP Restrictions page appears.

2. Select the option for the type of IP address you are adding, then complete the associated fields for that address type.

   Note: If you do not know the IP address but you have a domain name for the address, click the DNS Lookup button. This button opens the DNS Lookup page. Enter a fully qualified host name in the Host Name field and click OK.

   - Single Host--specifies a single IP address that hosts the browser. If you specify a single IP address, users can access the Service Provider only from the specified IP address.
   - Host Name--specifies a web server using its host name. If you specify a host name, the Service Provider is only accessible to users from the specified host.

- Subnet Mask--specifies a subnet mask for a web server. If you specify a subnet mask, the Service Provider is only accessible to users from the specified subnet mask. If you select this button, the Add An Address and Subnet Mask dialog opens. Use the Left and Right arrow buttons, or click and drag the slider bar to select a subnet mask.
- Range--specifies IP address range. If you specify a range of IP addresses, the Service Provider only permits users from one of the IP addresses in the range of addresses. Enter a starting (FROM) and ending (TO) addresses to determine the range.

3. Click OK to save your configuration.

Identify a Proxy Server (optional)

If your network has a proxy server between the client and the system with Federation Web Services, specify the protocol and authority portions of the URL. The syntax is *protocol*:*authority*.

protocol

http: or https:

authority

//*host.domain.com* or //*host.domain.com:port*

Example: http://example.ca.com.

To identify a proxy server

1. Begin at the General step of the configuration wizard.

   In the Advanced section of the page, enter the URL in the Server field.

2. Click Submit.

Select Users for Which Assertions are Generated

As part of the configuration at the asserting party, include a list of users and groups for which the Assertion Generator generates SAML assertions. The asserting party is either a SAML 1.x Producer, a SAML 2.0 Identity Provider, or a WS Federation Account Partner.

You can only add users and groups from directories that are in an affiliate domain.

To specify users and groups for federated transactions

1. Navigate to the Users settings for the partner you are configuring.

   The User Directories page displays entries for each user directory for the policy domain.

2. Add users or groups from the user directory to the policy.

   In each user directory table, you can select Add Members, Add Entry, Add All. Depending on which method you select, a dialog opens enabling you to add users.

   - 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.
   - If you select Add Entry, select users by manual entry (see page 125) in the User Directory Search Expression Edit dialog.

   Edit or delete a user or group by clicking the right arrow (>) or minus sign (-), respectively.

3. Select individual users, user groups, or both using whatever method and click OK.

   The User Directories page reopens and lists the new users in the user directory table.

More information:

Exclude a User or Group from Access to a Resource (see page 124)
Allow Nested Groups Access to Resources (see page 125)
Add Users by Manual Entry (see page 125)

Exclude a User or Group from Access to a Resource

You can exclude users or groups of users from obtaining an assertion.

Follow these steps:

1. Navigate to the User settings.
2. Select a user or group from the list for a particular user directory.
3. Click Exclude to exclude the selected user or group.

   The selection is reflected in the Administrative UI.

4. Click OK to save your changes.

Allow Nested Groups Access to Resources

LDAP user directories can contain groups that have subgroups. In complex directories, groups nesting in a hierarchy of other groups is one way to organize large amounts of user information.

If you enable a search for users in nested groups, any nested group is searched for the requested user record. If you do not enable nested groups, the Policy Server only searches the group you specify.

To enable searching in nested groups

1. Navigate to the Users settings.

   If the associated affiliate domain contains more than one user directory, each user directory appears in its own section.

2. Select the Allow Nested Groups check box to enable searching within nested groups.

Add Users by Manual Entry

When you specify users for assertion generation, one of the options is to identify users by manual entry.

Follow these steps:

1. Navigate to the Users settings for the partner you are configuring.

   If the affiliate domain contains more than one user directory, all the directories appear on the User Directories page.

2. Click Add Entry.

   The User Directory Search Express Edit page displays.

3. Select the search option then complete the fields for that search option.

   Where to Search

   For LDAP directories, select an option from the drop-down list:

   Validate DN

   LDAP search locates this DN in the directory.

   Search Users

   LDAP search is limited to matches in user entries.

Search Groups

LDAP search is limited to matches in group entries.

Search Organizations

LDAP search is limited to matches in organization entries.

Search Any Entry

LDAP search is limited to matches in user, group, and organization entries.

- For Microsoft SQL Server, Oracle and WinNT directories, you can enter a user name in the Manual Entry field.
- For a Microsoft SQL Server or Oracle, you can enter a SQL query instead. For example:

  SELECT NAME FROM EMPLOYEE WHERE JOB ='MGR';

  The Policy Server performs the query as the database user specified in the Username field of the Credentials and Connection tab for the user directory. When constructing the SQL statement for the Manual Entry field, be familiar with the database schema for the user directory. For example, if you are using the SmSampleUsers schema and you want to add specific users, select a user entry from the SmUser table.
- For an LDAP directory, enter **all** in the Manual Entry field to add all directory entries.

4. Click OK to save your changes.

Specify a Name ID for a SAML 2.0 Assertion

A name ID names a user in an assertion in a unique way. The name ID is added to the assertion sent to the Service Provider.

The format of the name ID establishes the type of content that is used for the ID. For example, the format can be the User DN, in which case the content would be a uid.

You can encrypt a Name ID. However, for single sign-on with the artifact binding, encrypting a NameID with other data in an assertion increases the size of the assertion.

Note: The NameID is required in an assertion.

To configure a name ID

1. Begin at the Name IDs step in the configuration wizard.
2. Select the Name ID Format.

   For a description of each format, see the *OASIS Security Assertion Markup Language (SAML) V2.0* specification.

3. Select the Name ID Type from the following options:
   - Static value
   - User attribute
   - DN attribute (with or without nested groups)

   The contents of the Name ID Fields section change according to the Name ID Type selected.
4. Complete the fields for the selected Name ID Type.

Note: If you configure Name IDs, do not select an affiliation in the SAML Affiliation field. Name IDs and affiliations are mutually exclusive.

More Information:

Encrypt a NameID and an Assertion (see page 228)

Customize a SAML Assertion Response (optional)

You can modify the assertion content using an assertion generator plug-in. The plug-in enables you to customize the content of an assertion using the business agreements between you and your partners and vendors. One plug-in is allowed for each partner.

The steps to configure an assertion generator plug-in are:

1. Install the CA SiteMinder® SDK, if you have not done so already.
2. Implement the AssertionGeneratorPlugin.java interface, which is part of the SDK.
3. Deploy your assertion generator plug-in implementation class.
4. Enable the assertion generator plug-in parameters in the Administrative UI.

Additional information about the Assertion Generator plug-in can be found as follows:

- Reference information (method signatures, parameters, return values, data types), and also the new constructor for UserContext class, are in the *Javadoc Reference*. Refer to the AssertionGeneratorPlugin interface in the Javadoc.
- Overview and conceptual information for authentication and authorization APIs is in the *CA SiteMinder® Programming Guide for Java*.

Implement the AssertionGeneratorPlugin Interface

The first step in creating a custom assertion generator plug-in is to implement the AssertionGeneratorPlugin interface.

Follow these steps:

1. Provide a public default constructor method that contains no parameters.
2. Provide code so that the implementation is stateless. Many threads must be able to use a single plug-in class.
3. Implement methods in the interface to satisfy your requirements.

   The implementation must include a call to the customizeAssertion methods. You can overwrite the existing implementations. See the following sample classes for examples:

   SAML 1.x/WS-Federation

   AssertionSample.java

   SAML 2.0

   SAML2AssertionSample.java

   The sample classes are located in the directory /sdk/samples/assertiongeneratorplugin.

   The contents of the parameter string that your implementation passes into the customizeAssertion method is the responsibility of the custom object.

Deploy the Assertion Generator Plug-in

After you have coded your implementation class for the AssertionGeneratorPlugin interface, compile it and verify that CA SiteMinder® can find your executable file.

To deploy the assertion generator plug-in

1. Compile the assertion plug-in Java file.

   Compilation requires the following .jar files, which are installed with the Policy Server:

   - *policy_server_home*/bin/jars/SmJavaApi.jar
   - *policy_server_home*/bin/thirdparty/xercesImpl.jar
   - *policy_server_home*/bin/endorsed/xalan.jar

2. In the JVMOptions.txt file, modify the -Djava.class.path value so it includes the classpath for the plug-in. This modification enables the plug-in to be loaded with the modified classpath. Locate the JVMOptions.txt file in the directory *installation_home*\siteminder\config.

   Note: Do not modify the classpath for xercesImpl.jar, xalan.jar, or SMJavaApi.jar.

3. Enable the plug-in.

Enable the Assertion Generator Plug-in

After writing an assertion generator plug-in and compiling it, enable the plug-in by configuring settings in the Administrative UI. The UI parameters let CA SiteMinder® know where to find the plug-in.

Do not configure the plug-in settings until you deploy the plug-in (see page 143).

Follow these steps:

1. Log on to the Administrative UI.
2. Click Federation, Legacy Federation, SAML Service Providers.
3. Select an existing Service Provider entry or create one.
4. Navigate to the General settings.
5. In the Assertion Generator Plug-in section, complete the following fields:

   Java Class Name

   Specify a Java class name for an existing plug-in

   The plug-in class can parse and modify the assertion, and then return the result to the Assertion Generator for final processing.

   Only one plug-in is allowed for each Service Provider. For example, com.mycompany.assertiongenerator.AssertionSample

   Parameter

   (Optional) Specify a string of parameters that is passed to the plug-in specified in the Java Class Name field.

   Note: Instead of enabling the assertion plug-in through the Administrative UI, you can use the Policy Management API (C or Perl) to integrate the plug-in. For more information, see the *CA SiteMinder® Programming Guide for C* or the *CA SiteMinder® Programming Guide for Java*.

6. Restart the Policy Server.

   Restarting the Policy Server ensures that the latest version of the assertion plug-in is picked up after being recompiled.

Customize the Assertion with Attributes from a Web Application

You can use an assertion generator plug-in to add web application attributes tp an assertion. This is another way to customize the assertion.

To include web application attributes in an assertion

1. Compile the assertion plug-in Java file.

   Compilation requires the following .jar files, which are installed with the Policy Server:

   - *policy_server_home*/bin/jars/SmJavaApi.jar
   - *policy_server_home*/bin/thirdparty/xercesImpl.jar
   - *policy_server_home*/bin/endorsed/xalan.jar

2. In the JVMOptions.txt file, modify the -Djava.class.path value so it includes the classpath for the plug-in. This modification enables the plug-in to be loaded with the modified classpath. Locate the JVMOptions.txt file in the directory *installation_home*\siteminder\config.

   Note: Do not modify the classpath for xercesImpl.jar, xalan.jar, or SMJavaApi.jar.

3. Configure a sample plug-in.

   An APIContext class in the SMJavaAPI has a new method, getAttrMap(), which returns a map object containing the attributes from the web application included in the assertion. In the SiteMinder SDK, there are two sample Assertion Generator plug-ins that show how to use this map object:

   - SAML2AppAttrPlugin.java (SAML 2.0)
   - WSFedAppAttrPlugin.java (WS-Federation)

   These samples are located in the directory sdk/samples/assertiongeneratorplugin. They enable the Assertion Generator to add attributes from a web application to an assertion.

4. Log in to the Administrative UI.
5. Select Federation, Legacy Federation, SAML Service Providers or Resource Partners.
6. Select an exiting entry or create one.
7. Navigate to the General settings.

8. In the Assertion Generator Plug-in section, complete the following fields:

   Java Class Name

   Names the Java class for the plug-in. For example, the sample classes included with the CA SiteMinder® SDK are:

   - com.ca.assertiongenerator.SAML2AppAttrPlugin

     (SAML 2.0)

   - com.ca.assertiongenerator.WSFedAppAttrPlugin

     (WS-Federation)

   Parameter

   Specify a string of parameters that is passed to the plug-in specified in the Java Class Name field. These parameters would be the attributes that you want to include in the assertion.

   Note: Instead of configuring the settings through the Administrative UI, you can use the Policy Management API (C or Perl) to integrate the plug-in. For instructions, see the *CA SiteMinder® Programming Guide for C* or the *CA SiteMinder® Programming Guide for Java*.

9. Restart the Policy Server.

   Restarting the Policy Server verifies that the latest version of the assertion plug-in is picked up after being recompiled.

Configure Single Sign-on for SAML 2.0

Part of the single sign-on configuration is to determine how the Identity Provider delivers an assertion to a Service Provider.

Follow these steps:

1. In the Administrative UI, navigate to the SAML Profiles settings for a Service Provider object.
2. Complete the fields in the SSO section of the page. Select the SAML bindings to use for communication.

   For the HTTP-Artifact binding, configure the back channel settings so the user has access to the protected Artifact Resolution Service. The back channel settings are on the Attributes step in the configuration wizard.
3. Click Submit to save your changes.

More information:

Configure the Authentication Scheme that Protects the Artifact Service (see page 133)

Assertion Validity for Single Sign-on

For single sign-on, the values of the Skew Time and the Validity Duration determine how CA SiteMinder® calculates the total time that an assertion is valid. CA SiteMinder® applies the skew time to the generation and consumption of assertions.

Note: In this description, the asserting party is the SAML 1.x Producer, SAML 2.0 Identity Provider, or WS-Federation Account Partner. The relying party is the SAML 1.x Consumer, the SAML 2.0 Service Provider, or the WS-Federation Resource Partner.

In the assertion document, the NotBefore and NotOnOrAfter values represent the beginning and end of the validity interval.

At the asserting party, CA SiteMinder® sets the assertion validity. The validity interval is the system time when the assertion is generated. CA SiteMinder® sets the IssueInstant value in the assertion using this time then subtracts the skew time value from the IssueInstant value. The resulting time is the NotBefore value.

NotBefore=IssueInstant - Skew Time

To determine the end of the validity interval, CA SiteMinder® 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, CA SiteMinder® performs the same calculations as it does at the asserting party to determine if the assertion it receives is valid.

Calculating Assertion Validity with CA SiteMinder® at Both Sides of the Partnership

If CA SiteMinder® is at both sides of a partnership, the assertion validity is the sum of the validity duration plus two times the skew time. The equation is:

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

The initial part of the equation (2 x Skew Time + Validity Duration) represents the beginning and end of 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 legacy federation, the Validity Duration is only set at the asserting party.

Example

Asserting Party

The values at the asserting party are as follows:

- IssueInstant=5:00PM
- Validity Duration=60 seconds
- Skew Time = 60 seconds
- NotBefore = 4:59PM
- NotOnOrAfter=5:02PM

Relying Party

The relying party uses the NotBefore and NotOnOrAfter values from the assertion and applies its skew time to those values. This formula is how the relying party calculates new NotBefore and NotOnOrAfter values.

- Skew Time = 180 seconds (3 minutes)
- NotBefore = 4:56PM
- NotOnOrAfter=5:05PM

Assertion Validity Window

Using the values in this example, the calculation for the total assertion validity window is:

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

Define Indexed Endpoints for Different Single Sign-on Bindings

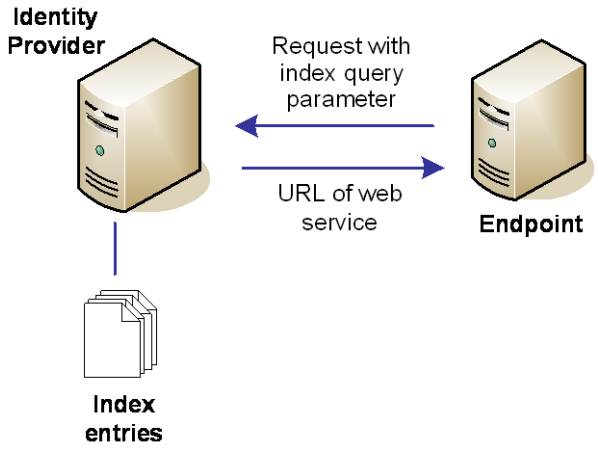
You can configure indexed endpoints for federated communication. An indexed endpoint is the site where assertions are consumed. In the context of CA SiteMinder®, this endpoint is the Service Provider where the Assertion Consumer Service resides.

Each endpoint you configure is assigned a unique index value, instead of a single, explicit reference to an Assertion Consumer Service URL. The assigned index is added to the assertion request that the Service Provider sends to the Identity Provider.

You can configure indexed endpoints for a CA SiteMinder® Service Provider that has a federated relationship with a third-party Identity Provider that supports indexed endpoints. You can also configure the different protocol bindings (artifact, POST) for the Assertion Consumer Service by assigning more than one endpoint to the service.

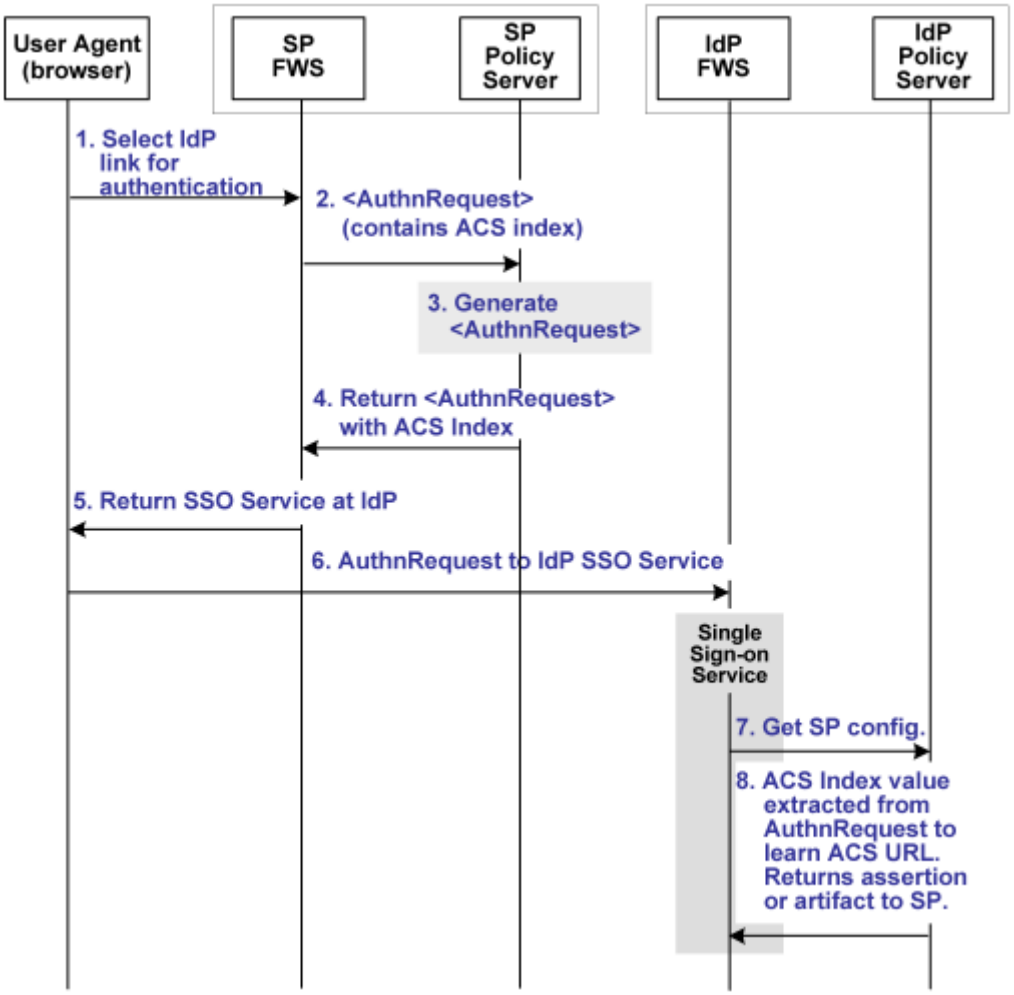
Note: If your network contains different CA SiteMinder® versions, you cannot configure indexed endpoints. For example, you cannot configure indexed endpoints if the Service Provider is r12.0 SP 2 and the Identity Provider is r12.0 SP3. Configure only one Assertion Consumer Service for both HTTP bindings.

The following figure shows a network that benefits from indexed endpoints.



Indexed Endpoints Flow Diagram

The following illustration shows how single sign-on works using an indexed endpoint.



Note: The Web Agent Option Pack or the SPS federation gateway can provide the FWS functionality.

Using indexed endpoints, the sequence of events is as follows:

1. The user selects a link to authenticate with a specific IdP. The link contains the IdP ID and AssertionConsumerServiceIndex query parameters index as query parameters because the index feature is enabled.
2. The SP Federation Web Services (FWS) application asks for an AuthnRequest from its local Policy Server. The request that it sends includes the IdP ID and optionally, the AssertionConsumerServiceIndex and ForceAuthn query parameters.

   A protocol binding is not part of the request because the ACS Index and the Protocol Binding parameters are mutually exclusive. The AssertionConsumerServiceIndex is already associated with a binding so there is no need to specify a Protocol Binding value. If the protocol binding and the AssertionConsumerServiceIndex are passed as query parameters, the local Policy Server responds with an error denying the request.
3. The AuthnRequest service extracts the IdP information from the SP Policy Server and generates the AuthnRequest message, which includes the AssertionConsumerServiceIndex. Because the AssertionConsumerServiceIndex is one of the query parameters, its value is verified against the IdP from an IdP descriptor document. This document is previously sent from the IdP to the SP.

   The AuthnRequest service reacts as follows:

   - If the index for the artifact binding is set in the IdP metadata, this index is compared to the AssertionConsumerServiceIndex value. If the values match, the index value remains part of the AuthnRequest. If the index values do not match, the IdP metadata is verified. The AssertionConsumerServiceIndex must correspond to the POST binding.
   - If the index corresponding to the HTTP-POST binding, this index value is again compared with the AssertionConsumerServiceIndex in the AuthnRequest. If the value of the AssertionConsumerServiceIndex parameter does not match the POST binding, the AuthnRequest service generates an error. The error states that the AssertionConsumerServiceIndex does not match the index in the IdP metadata.
4. Assuming that the IdP metadata index and AssertionConsumerServiceIndex values match, the SP Policy Server generates the AuthnRequest.
5. The SP Policy Server returns the AuthnRequest in an HTTP-redirect binding.
6. The the SP FWS application redirects the AuthnRequest to the single sign-on service at the IdP. The SP knows the URL of the single sign-on service because the URL is part of the configuration information in the AuthnRequest.

7. The browser requests the single sign-on service.
8. The single sign-on service extracts the AssertionConsumerServiceIndex value from the AuthnRequest. The service determines the Assertion Consumer Service URL using the AssertionConsumerServiceIndex. If the Index is not in the metadata, the service generates an error. The error message indicates that an invalid AssertionConsumerServiceIndex is in the AuthnRequest message.

   The Assertion Consumer URL to send the assertion or artifact to the SP, depending on the single sign-on profile in use.

   Note: If the AssertionConsumerServiceIndex parameter is not in the AuthnRequest, the value of the Assertion Consumer Service and the corresponding binding are used by default.

Configure Indexed Endpoints for the Assertion Consumer Service

The single sign-on service extracts an ACS Index value from an AuthnRequest. The service compares the index value to its list of index entries and determines the Assertion Consumer Service URL associated with that index value. The single sign-on service then knows where to send the assertion or artifact, depending on the binding that is associated with the index value.

To configure index entries at the Identity Provider

1. Log in to the Administrative UI.
2. Select a Service Provider entry to modify or create one.
3. Navigate to the SAML Profiles page.
4. In the SSO section of the page, click the ellipses button at the end of the Assertion Consumer Service field.

   The Assertion Consumer Services page opens.
5. Click Add.

   The Add Assertion Consumer Service page opens.
6. Complete the fields on the page.

   You can assign different index values to the same Assertion Consumer Service URL.
7. Click OK to save your changes.

Note: Remember to configure index entries in the SAML 2.0 authentication scheme at the Service Provider.

Enforce the Authentication Scheme Protection Level for SSO

When a user requests a federated resource, they must have a CA SiteMinder® session. If a user does not have a session, the user is redirected to the Authentication URL to establish a session. The authentication scheme protecting the Authentication URL is configured with a particular protection level. This protection level must be the same or greater than the authentication level you configure for the SAML Service Provider configuration.

If the protection level for the Authentication URL is less than the Authentication Level set in the Administrative UI, the Policy Server does not generate an assertion.

Determine Digital Signing Options

Federation uses a private key/certificate pair to perform various digital signing tasks for federated communication. The private key/certificate pair can sign the following messages:

- Assertions
- SAML responses
- Artifact responses
- Single logout requests and responses

  For single logout, the side that initiates the logout signs the request. The side receiving the request validates the signature. Conversely, the receiving side must sign the SLO response and the initiator must validate the response signature.
- Attribute responses (for authorizations that are based on user attributes)

The partner responsible for signing gives the certificate (public key) associated with the private key to the partner that verifies the signature. This exchange is done in an independent communication before any transactions occur.

When an IdP sends an assertion to an SP, it includes the certificate in the assertion, by default. However, the SP uses the certificate that it stores at its site to verify the signature.

The configuration options for digital signing include:

- Signature alias
- Signature algorithm (RSAwithSHA1 or RSAwithSHA256)
- HTTP-Artifact assertion, SAML response, and artifact response options
- HTTP-POST assertion and SAML response options

To specify signing options from the General or SSO tab

1. Modify an existing SAML Service Provider object or create one.
2. Navigate to SAML Profiles.
3. In the Signing Options section of the dialog, complete the fields.
4. Click Submit.

Enhanced Client or Proxy Profile Overview

The Enhanced Client or Proxy Profile (ECP) is an application for single sign-on. An enhanced client is a browser or some other user agent that supports the ECP functionality. An enhanced proxy is an HTTP proxy, such as a Wireless Access Protocol proxy for a wireless device.

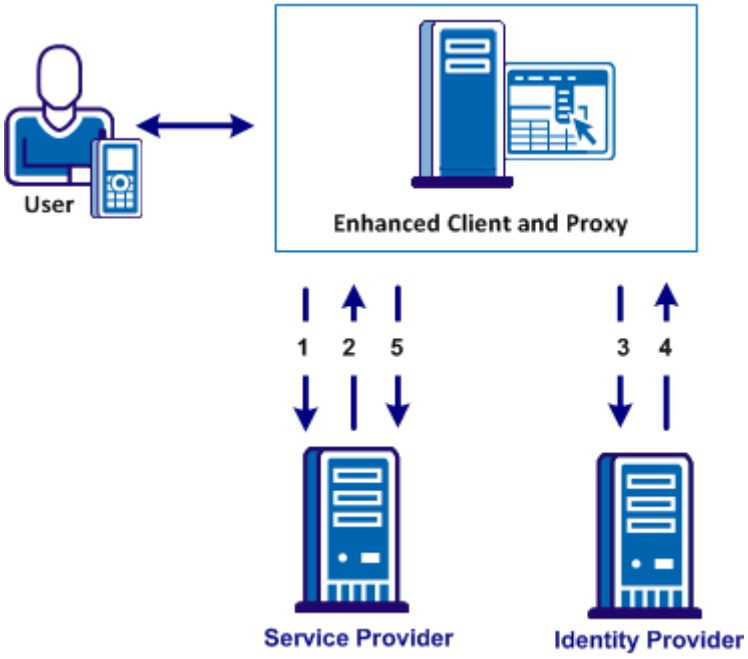
The ECP profile enables single sign-on when the Identity Provider and Service Provider cannot communicate directly. The ECP acts as the intermediary between the Service Provider and the Identity Provider.

In addition to acting as an intermediary, the ECP profile is useful in the following situations:

- For a Service Provider that expects to service enhanced clients or proxies that require this profile.
- When a proxy server is in use, such as a wireless access protocol (WAP) gateway in front of a mobile device with limited functionality.

You are responsible for obtaining or developing an ECP application. The Policy Server only processes ECP requests and only responds to the ECP application in keeping with the SAML requirements.

The flow of the ECP profile is shown in the following illustration.



In an ECP communication, a user requests access to an application, for example, from a mobile phone. The application resides at the Service Provider and the identity information for the user resides at the Identity Provider. The Service Provider and Identity Provider do not communicate directly.

The flow of the ECP call is as follows:

1. The ECP application forwards a reverse SOAP (PAOS) request to the Service Provider. The Identity Provider is not directly accessible by the Service Provider.

   The ECP entity is always directly accessible, unlike the Identity Provider.
2. The Service Provider sends an AuthnRequest back to the ECP application.
3. The ECP application processes and modifies the AuthnRequest and sends it on to the Identity Provider.
4. The Identity Provider processes the request and returns a SOAP response to the ECP application. This response includes the assertion.
5. The ECP application passes a signed PAOS response back to the Service Provider.

Single sign-on proceeds and the user gains access to the application.

Configure ECP at the Identity Provider

To configure ECP, enable the feature at the Identity Provider and the Service Provider. The following procedure is for a CA SiteMinder® Identity Provider.

Follow these steps:

1. Log in to the Administrative UI at the Identity Provider.
2. Navigate to the SAML Profiles tab for the SAML Service Provider you want to modify.
3. Complete the required single sign-on configuration settings in the dialog.
4. Select the Enhanced Client and Proxy Profile check box in the SSO section.
5. Click Submit.

The Identity Provider can now process ECP calls.

Your federated partner must also enable ECP. For CA SiteMinder®, enable ECP at the SAML 2.0 authentication scheme (see page 244).

Note: A single SAML Service Provider object can handle artifact, POST, SOAP, and PAOS bindings for single sign-on requests. SOAP and PAOS are the bindings for the ECP profile. The Identity Provider and Service Provider determine the binding being used based on the parameters in a request.

Create a User Identifier by Enabling Allow/Create

The SAML 2.0 Allow/Create feature lets an Identity Provider create a user identifier at the request of the Service Provider. For this feature to work, the Service Provider request must include the Allow/Create attribute. Additionally, an administrator must configure the Identity Provider to create an identifier. The Identity Provider generates a unique value that becomes part of the NameID, which is in the assertion returned to the Service Provider.

When the Service Provider receives the assertion, the SAML 2.0 authentication scheme processes the response. The scheme then looks up the user in its local user store. Assuming the user record is located, the user is granted access.

To enable the creation of a new user identifier

1. Log in to the Administrative UI.
2. Navigate to the SAML Profile settings for a Service Provider object.
3. In the SSO section of the page, select Allow Creation of New Identifier.
4. Click Submit.

Ignore the Authentication Context from the SP

Exchanging information about the user authentication context provides each side of a federated partnership a way of communicating about the authentication process.

If a Service Provider requests the authentication context in a request to the Identity Provider, you can configure the Identity Provider to ignore the context.

To ignore the authentication context

1. Navigate to the General settings for a Service Provider object.
2. In the Advanced SSO Configuration section, select the Ignore Requested AuthnContext.
3. Click Submit.

Configure Assertions for One-time Use

In compliance with the SAML 2.0 specification, the Policy Server can enforce the one time use of an assertion. By generating an assertion for one-time use, the relying party knows not to retain the assertion for future transactions. Reusing an assertion beyond its validity results in authentication decisions that is based on out-of-date identity information.

To configure an assertion for one time use

1. Navigate to the General settings for a Service Provider object.
2. In the Advanced SSO Configuration section, select Set OneTimeUse Condition.
3. Click Submit.

HTTP Error Handling at the IdP

Assertion-based single sign-on can fail at the Identity Provider for various reasons. If an HTTP error occurs, the user can be redirected to different applications (URLs) for further processing. Redirection to a customized error page can take place only when the Identity Provider has the necessary information about the Service Provider. If the information is not available, only the HTTP error code is returned to the browser. No redirection occurs.

You can configure redirect URLs for HTTP handling, but they are not required.

To configure optional redirect URLs for error handling

1. Navigate to the General settings.
2. In the Advanced SSO Configuration section, select the URL you want to enable then enter the URL. You can specify URLs for one or more of the following errors:
   - Enable Server Error URL
   - Enable Invalid Request URL
   - Enable Unauthorized Access URL
3. Select one of the following options for the Mode:
   - 302 No Data
   - HTTP POST
4. Click Submit.

Note: These redirect URLs can be used with the CA SiteMinder® Assertion Consumer Plug-in for further assertion processing. If an assertion request fails, the plug-in can send the user to one of the redirect URLs you specify.

Customize the Session Duration in the Assertion

When the Policy Server IdP sends an assertion, by default it includes the SessionNotOnOrAfter parameter in the Authentication statement of the assertion. A third-party SP can use the value of the SessionNotOnOrAfter to set its own timeout values. The timeout values determine when a user session becomes invalid, which sends the user to reauthenticate at the IdP.

Important! If the Policy Server is acting as an SP, it ignores the SessionNotOnOrAfter value. Instead, the SP sets session timeouts based on the realm timeout that corresponds to the configured SAML authentication scheme that protects the target resource.

The SessionNotOnOrAfter parameter is different than the NotOnOrAfter parameter used to determine assertion validity and skew time.

To customize the SessionNotOnOrAfter parameter

1. Log on to the UI.
2. Select the Service Provider entry that you want to modify.
3. Navigate to the Advanced tab.
4. Select the Customize Validity duration in the Advanced SSO Configuration section of the dialog.

   The Customize Validity duration dialog displays.

5. Select a value for the SP Session Validity Duration. The value that you enter is the value of the SessionNotOnOrAfter parameter in the assertion.

   The options are:

   Use Assertion Validity

   Calculates the SessionNotOnOrAfter value that is based on the assertion validity duration.

   Omit

   Instructs the IdP not to include the SessionNotOnOrAfter parameter in the assertion.

   IDP Session

   Calculates the SessionNotOnOrAfter value that is based on the IdP session timeout. The timeout is configured in the IdP realm for the authentication URL. Using this option can synchronize the IdP and SP session timeout values.

   Custom

   Lets you specify a custom value for the SessionNotOnOrAfter parameter in the assertion. If you select this option, enter a time in the Customize Assertion Session Duration field.

6. Click OK to save the changes.

Grant Access to the Service for Assertion Retrieval (Artifact SSO)

For HTTP-Artifact single sign-on, the relying party needs permission to access the policy that protects the FWS service for obtaining assertions.

To grant access:

- Add the Web Agent that protects the FWS application to the agent group FederationWebServicesAgentGroup.
- Add relying partners as users (see page 131) who are permitted to access the specific service.

  Other than adding users to a given policy, all other policy objects are set up automatically.

Add a Web Agent to the Federation Agent Group

Add the Web Agent that protects the FWS application to the Agent group FederationWebServicesAgentGroup.

- For ServletExec, this Agent is on the web server where the Web Agent Option Pack is installed.
- For an application server, such as WebLogic or JBOSS, this Web Agent is installed where the application server proxy is installed. The Web Agent Option Pack can be on a different system.

Follow these steps:

1. Log in to the Administrative UI.
2. Click Infrastructure, Agent, Agents.
3. Click Create Agent.
4. Specify the name of the Web Agent in your deployment. Click Submit.
5. Click Infrastructure, Agent, Agent Groups.
6. Select the FederationWebServicesAgentGroup entry.
7. Click Add/Remove and the Agent Group Members dialog opens.
8. Move the web agent from the Available Members list to the Selected Members list.
9. Click OK to return to the Agent Groups dialog.
10. Click Submit then click Close to return to the main page.

Add Relying Partners to the FWS Policy for Obtaining Assertions

If you are using HTTP-Artifact binding for single sign-on, the relying party in the partnership needs permission to access the assertion retrieval service. CA SiteMinder® protects the SAML 1.x and 2.0 retrieval services with a policy.

When you install the Policy Server, the FederationWebServicesDomain is installed by default. This domain includes the following policies for the service from which CA SiteMinder® retrieves assertions:

SAML 1.x

FederationWSAssertionRetrievalServicePolicy

SAML 2.0

SAML2FWSArtifactResolutionServicePolicy

Note: WS-Federation does not use the HTTP-Artifact profile. Therefore, this procedure does not apply to Resource Providers.

Grant access for these policies to any relevant relying partners.

Follow these steps:

1. In the Administrative UI, navigate to Policies, Domain, Domain Policies.

   A list of domain policies displays.

2. Select the policy for the SAML profile:

   SAML 1.x

   FederationWSAssertionRetrievalServicePolicy

   SAML 2.0

   SAML2FWSArtifactResolutionServicePolicy

   The Domain Policies page opens.

3. Click Modify to change the policy.

4. Select the Users tab.

5. In the dialog for the appropriate user directory, click Add Members:

   SAML 1.x

   FederationWSCustomUserStore

   SAML 2.0

   SAML2FederationCustomUserStore

   The User/Groups page opens.

   The affiliate domain that you previously configured is listed in the Users/Groups dialog. For example, if the affiliate domain is named fedpartners, the entry is **affiliate:fedpartners**.

6. Select the check box next to the affiliate domain with the partners that require access to the service. Click OK.

   You return to the User Directories list.

7. Click Submit.

   You return to the policies list.

Configure the Authentication Scheme that Protects the Artifact Service

For the HTTP-Artifact profile, the assertion retrieval service (SAML 1.x) and the artifact resolution service (SAML 2.0) retrieve the assertion at the asserting party. When these services send an assertion response to the relying party, they do so over a secure back channel. We strongly recommend that you protect these services and the communication across the back channel against unauthorized access.

Note: WS-Federation does not support the HTTP-Artifact profile.

To protect these services, specify an authentication scheme for the realm that contains the service at the asserting party. The authentication scheme dictates the type of credentials that the consuming service at the relying party must provide to access the relevant service across the back channel.

You can select one of the following authentication schemes:

- Basic (see page 134)
- Basic over SSL (see page 134)
- X.509 client certificate (see page 135)

Basic Authentication to Protect the Service that Retrieves Assertions

For HTTP-Artifact single sign-on, the asserting party sends the assertion across a secure back channel to the relying party. For basic authentication, configure a password to access to the service that resolves the artifact and retrieves the assertion. The service then sends the assertion across the back channel to the relying party.

You can use Basic authentication with SSL is enabled; however, SSL is not required.

Note: The password is only relevant if you use Basic or Basic over SSL as the authentication method across the back channel.

Follow these steps: for the SAML 1.x Assertion Retrieval Service

1. Log in to the Administrative UI.
2. Navigate to the General settings for the producer.
3. Enter a value for the following fields:
   - Password
   - Confirm Password
4. Click Submit to save the changes.

Follow these steps: for the SAML 2.0 Artifact Resolution Service

1. Log in to the Administrative UI.
2. Navigate to the Attribute settings for the Identity Provider.
3. In the Backchannel section, enter a value for the following fields:
   - Password
   - Confirm Password
4. Click Submit to save the changes.

Basic over SSL to Protect the Service that Retrieves Assertions

You can protect the assertion retrieval service (SAML 1.x) or the artifact resolution service (SAML 2.0) with a Basic over SSL authentication scheme. At the asserting party, a set of default policies to protect the service is already configured when you install the Policy Server.

The only configuration that is required is to enable SSL at each partner. No other configuration is required at the asserting or relying party. At the relying party, you can use one of the default root Certificate Authorities (CAs) in the certificate data store to establish an SSL connection. To use your own root CA instead of a default CA, import the CA certificate into the data store.

If you use Basic over SSL authentication scheme, all endpoint URLs have to use SSL communication. This means that the URLs must begin with **https://**. Endpoint URLs locate the various SAML services on a server, such as single sign-on, single logout, the Assertion Consumer Service, Artifact Resolution Service (SAML 2.0), and the Assertion Retrieval Service (SAML 1.x).

Client Certificate Auth to Protect the Service that Retrieves Assertion

You can protect the Assertion Retrieval Service (SAML 1.x) and the Artifact Resolution Service (SAML 2.0) with a client certificate authentication scheme. If the asserting party is configured to require client certificate authentication, the relying party makes a connection back to the asserting party and attempts to present a client certificate.

To use a client certificate authentication scheme:

1. Create a policy at the asserting party to protect the relevant service. This policy uses the client certificate authentication scheme.
2. Enable client certificate authentication for the back channel configuration at the relying party.
3. Enable SSL at each side of the partnership.

If you use Client Cert authentication, all endpoint URLs have to use SSL communication. Therefore, URLs must begin with **https://**. Endpoint URLs locate the various SAML services on a server, such as single sign-on, single logout, the Assertion Consumer Service, Artifact Resolution Service (SAML 2.0), and the Assertion Retrieval Service (SAML 1.x).

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.

Create the Policy to Protect the Retrieval Service

Create the policy at the asserting party to protect the service from which the asserting party retrieves the assertion.

Follow these steps:

1. For each affiliate requesting assertions, add a separate entry to a user directory. Create a user directory or use an existing directory.

   In the user record, enter the same value that is specified in the Name field of the affiliate general settings in the Administrative UI. For example, if Company A is the value of the Name field for the affiliate, the user directory entry is:

   uid=CompanyA, ou=Development,o=CA

   The Policy Server maps the subject DN value of the affiliate client certificate to this directory entry.
2. Add the configured user directory to the FederationWebServicesDomain.
3. Create a certificate mapping entry.

   Map the Attribute Name to the user directory entry for the affiliate. The attribute represents the subject DN entry in the certificate for the affiliate. For example, you select CN as the Attribute Name, and this value represents the affiliate named cn=CompanyA,ou=Development,o=partner.

   Navigate to Infrastructure, Directory, Certificate Mappings for the mapping settings.

4. Configure an X509 Client Certificate authentication scheme.
5. Create a realm under the FederationWebServicesDomain containing the following entries:

   Name

   any_name

   Example: cert assertion retrieval

   Agent

   FederationWebServicesAgentGroup

   Resource Filter

   /affwebservices/certassertionretriever (SAML 1.x)

   /affwebservices/saml2certartifactresolution (SAML 2.0)

   Authentication Scheme

   Client certificate authentication scheme created in the previous step.
6. Create a rule under the cert assertion retriever realm containing the following information:

   Name

   any_name

   Example: cert assertion retrieval rule

   Resource

   *

   Web Agent Actions

   GET, POST, PUT
7. Create a Web Agent response header under the FederationWebServicesDomain.

   The assertion retrieval service uses this HTTP header to verify that the affiliate is the site retrieving the assertion.

   Create a response with the following values:

   Name

   any_name

   Attribute

   WebAgent-HTTP-Header-Variable

   Attribute Kind

   User Attribute

Variable Name

consumer_name

Attribute Name

Enter the use directory attribute that contains the affiliate name value.

Example: uid=CompanyA.

Based on the following entries, the Web Agent returns a response named HTTP_CONSUMER_NAME.

8. Create a policy under the FederationWebServicesDomain containing the following values:

Name

any_name

User

Add the users from the user directory created in previously in this procedure.

Rule

rule_created_earlier_in_this_procedure

Response

response_created_earlier_in_this_procedure

The policy to protect the artifact resolution service is complete.

At the relying party, the administrator has to enable client certificate authentication across the back channel that connects to the relevant assertion service:

SAML 1.x: Enable client certificate authentication (see page 168) for the Assertion Retrieval Service

SAML 2.0: Enable client certificate authentication (see page 263) for the Artifact Resolution Service

WebLogic Configuration Required for Back Channel Authentication

At the Identity Provider, the Web Agent Option Pack can be installed on a WebLogic 9.2.*x* application server. For basic authentication across the back channel to work with this server, modify the WebLogic config.xml file.

In the WebLogic config.xml file for the application domain, set the <enforce-valid-basic-auth-credentials> within the <security-configuration> element as follows:

```
<enforce-valid-basic-auth-credentials>false</enforce-valid-basic-auth-credentials>
```

Initiate Single Sign-on from the IdP or SP

To initiate single sign-on, the user can begin at the Identity Provider or the Service Provider. Configure links at each site or as part of applications to trigger single sign-on operation.

More information:

Identity Provider-initiated SSO (POST or artifact binding) (see page 211)
Service Provider-initiated SSO (POST or artifact binding) (see page 214)

Identity Provider-initiated SSO (POST or artifact binding)

If a user visits the Identity Provider before going to the Service Provider, the Identity Provider must generate an unsolicited response. To initiate an unsolicited response, create a hard-coded link that generates an HTTP Get request that includes a query parameter with the Service Provider ID. The Identity Provider generates an assertion response for this ID. The Federation Web Service application and the Assertion Generator must accept the GET request.

A user clicks the link you establish to initiate the unsolicited response.

To specify the use of artifact or POST profile in the unsolicited response, the syntax for the unsolicited response link is:

`http://idp_server:port/affwebservices/public/saml2sso?SPID=SP_ID&ProtocolBinding=URI_for_binding`

idp_server:port

Identifies the web server and port hosting the Web Agent Option Pack or SPS federation gateway.

SP_ID

Service Provider ID value. The entity ID is case-sensitive. Enter it exactly as it appears in the Administrative UI.

URI_for_binding

Identifies the URI of the POST or Artifact binding for the ProtocolBinding element. The SAML 2.0 specification defines this URI.

Also specify the binding in the SAML Service Provider properties for the unsolicited response to work.

Note the following information:

- If there is no ProtocolBinding parameter in the link and only one binding in the Service Provider properties, the Service Provider uses the one binding.
- If the artifact and POST bindings are enabled in the Service Provider properties, POST is the default. Therefore, if you want to *only* use artifact binding, include the ProtocolBinding query parameter in the link.
- If you configure indexed endpoints for the Assertion Consumer Services, the ProtocolBinding query parameter overrides the binding for the Assertion Consumer Service.

More information:

Unsolicited Response Query Parameters that the IdP Uses (see page 212)

Unsolicited Response Query Parameters that the IdP Uses

An unsolicited response that initiates single sign-on from the Policy Server IdP can include the following query parameters:

- SPID
- ProtocolBinding
- RelayState

SPID

(Required) Specifies the ID of the Service Provider where the Identity Provider sends the unsolicited response. The entity ID is case-sensitive. Enter it exactly as it appears in the Administrative UI.

ProtocolBinding

Specifies the ProtocolBinding element in the unsolicited response. This element specifies the protocol used when sending the assertion response to the Service Provider. If the Service Provider is not configured to support the specified protocol binding, the request fails.

Required Use of the ProtocolBinding Query Parameter

Using the ProtocolBinding parameter is required *only* if the artifact and POST bindings are enabled in the Service Provider properties. If both profiles are enabled, use the query parameter only to use artifact binding.

- The URI for the artifact binding from the SAML 2.0 specification is:

  urn:oasis:names:tc:SAML:2.0:bindings:HTTP-Artifact

- The URI for the POST binding from the SAML 2.0 specification is:

  urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST

  You do not need to set this parameter for HTTP-POST single sign-on.

Note: Do not HTTP-encode the query parameters.

Example: Unsolicited Response with ProtocolBinding

This link redirects the user to the Single Sign-on service. In this link is the Service Provider identity. The SPID query parameter indicates the identity. Additionally, the bindings query parameter indicates that the artifact binding is in use. After the user clicks this hard-coded link, they are redirected to the local Single Sign-on service.

```
http://idp-ca:82/affwebservices/public/saml2sso?SPID=http%3A%2F%2Ffedsrv.
acme.com
%2Fsmidp2for90&ProtocolBinding=urn:oasis:names:tc:SAML:2.0:bindings:HTTP-
Artifact
```

Optional Use of the ProtocolBinding Query Parameter

If you *do not* use the ProtocolBinding query parameter, the following conditions apply:

- If the ProtocolBinding is not specified in the unsolicited response, the profile for the Service Provider is used.
- Both profiles can be enabled for the Service Provider. If the ProtocolBinding is not in the unsolicited response, the Service Provider uses the POST profile by default.

Example: Unsolicited Response without ProtocolBinding

This link redirects the user to the Single Sign-on service. Included in this link is the Service Provider identity. The SPID query parameter indicates the identity. No ProtocolBinding query parameter exists. After the user clicks this hard-coded link, they are redirected to the local Single Sign-on service.

```
http://fedsrv.fedsite.com:82/affwebservices/public/saml2sso?SPID=
http%3A%2F%2Ffedsrv.acme.com%2Fsmidp2for90
```

RelayState

Specifies the target at the Service Provider. Use the RelayState query parameter to indicate the target destination; however, this method is optional. There can be a configuration mechanism at the Service Provider to indicate the target.

URL-encode the RelayState value.

Example

```
http://ca.sp.com:90/affwebservices/public/saml2authnrequest?ProviderID=
http%3A%2F%2Ffedsrv.acme.com%2Fsmidp2for90&
RelayState=http%3A%2F%2Fwww.spdemo.com%2Fapps%2Fapp.jsp
```

Service Provider-initiated SSO (POST or artifact binding)

A user can visit the Service Provider first and then go to an Identity Provider. Therefore, create an HTML page at the Service Provider containing hard-coded links to its AuthnRequest service. The links in the HTML page redirect the user to the Identity Provider for authentication. The links also indicate what is in the AuthnRequest.

The hard-coded link that the user selects must contain specific query parameters. These parameters are part of the HTTP GET request to the AuthnRequest service at the Service Provider.

Note: The page with these hard-coded links has to reside in an unprotected realm.

To specify the use of artifact or profile binding for the transaction, the syntax for the link is:

`http://`*`SP_server:port`*`/affwebservices/public/saml2authnrequest?ProviderID=`*`IdP_ID`*`&ProtocolBinding=`*`URI_of_binding`*

sp_server:port

Specifies the server and port number at the Service Provider hosting the Web Agent Option Pack or the SPS federation gateway.

IdP_ID

Specifies the identity that is assigned to the Identity Provider. The entity ID is case-sensitive. Enter it exactly as it appears in the Administrative UI.

URI_of_binding

Identifies the URI of the POST or Artifact binding for the ProtocolBinding element. The SAML 2.0 Specification defines this URI.

For the request to work, enable a binding for the SAML authentication scheme.

Note the following information:

- If you do not include the ProtocolBinding query parameter in the AuthnRequest, the default binding is the one defined for the authentication scheme. If you have both bindings defined in the authentication scheme, then no binding is passed in the AuthnRequest. As a result, the default binding at the Identity Provider is used.
- Artifact and POST can be enabled for the SAML authentication scheme. Include the ProtocolBinding query parameter in the link if you only want to use artifact binding.

AuthnRequest Query Parameters Used by a SiteMinder SP

A CA SiteMinder® Service Provider can use query parameters in the links to the AuthnRequest Service. The allowable query parameters are:

ProviderID (required)

ID of the Identity Provider where the AuthnRequest Service sends the AuthnRequest message. The entity ID is case-sensitive. Enter it exactly as it appears in the Administrative UI.

ProtocolBinding

Specifies the ProtocolBinding element in the AuthnRequest message. This element specifies the protocol that the Identity Provider uses to return the SAML response. If the specified Identity Provider is not configured to support the specified protocol binding, the request fails.

If you use this parameter in the AuthnRequest, you cannot include the AssertionConsumerServiceIndex parameter also. They are mutually exclusive.

Required Use of the ProtocolBinding Query Parameter

The artifact and POST binding can be enabled for an authentication scheme. If you want to use only the artifact binding, the ProtocolBinding parameter is required.

- The URI for the artifact binding from the SAML 2.0 specification is:

  urn:oasis:names:tc:SAML:2.0:bindings:HTTP-Artifact

- The URI for the POST binding as from the SAML 2.0 specification is:

  urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST

  You do not need to set this parameter for HTTP-POST single sign-on.

  Example: AuthnRequest Link with ProtocolBinding

  ```
  http://ca.sp.com:90/affwebservices/public/saml2authnrequest?ProviderID=
  http%3A%2F%2Ffedsrv.acme.com%2Fsmidp2for90&ProtocolBinding=urn:oasis:
  names:tc:SAML:2.0:bindings:HTTP-Artifact
  ```

  A user clicks the link at the Service Provider. The Federation Web Services application requests an AuthnRequest message from the local Policy Server.

Optional Use of ProtocolBinding

When you *do not* use the ProtocolBinding query parameter, the following conditions apply:

- If only one binding is enabled for the authentication scheme and the ProtocolBinding query parameter is not specified, the authentication scheme uses the enabled binding.
- If both bindings are enabled and the ProtocolBinding query parameter is not specified, POST binding is used as the default.

Note: Do not HTTP-encode the query parameters.

Example: AuthnRequest Link without ProtocolBinding

This sample link goes to the AuthnRequest service. The link specifies the Identity Provider in the ProviderID query parameter.

```
http://ca.sp.com:90/affwebservices/public/saml2authnrequest?ProviderID=
http%3A%2F%2Ffedsrv.acme.com%2Fsmidp2for90
```

A user clicks the link at the Service Provider. The Federation Web Services application requests for an AuthnRequest message from the local Policy Server.

ForceAuthn

Indicates whether the SP forces the Identity Provider to authenticate a user even if there is an existing security context for that user.

- If ForceAuthn=True in the AuthnRequest message, and a CA SiteMinder® session exists for a particular user, the IdP rechallenges the user for credentials. If the user successfully authenticates, the IdP includes the identity information from the existing session in the assertion. The IdP discards the session that it generated for reauthentication.

  Note: A user can try to reauthenticate with different credentials than the existing session. The IdP then compares the userDN and the user directory OID for the current and existing sessions. If the sessions are not for the same user, the IdP returns a SAML 2.0 response. The response indicates that the authentication has failed.

- If the SP sets ForceAuthn=True in the AuthnRequest message and there is no SiteMinder session, the SiteMinder IdP challenges the user for credentials. If the user successfully authenticates, a session is established.

Example

http://www.sp.demo:81/affwebservices/public/saml2authnrequest?ProviderID=idp.demo&ForceAuthn=yes

RelayState

Specifies the target at the Service Provider. You can use the RelayState query parameter to indicate the target destination, but this method is optional. Instead, you can specify the target configured in the SAML 2.0 authentication scheme. The authentication scheme also has an option to override the target with the RelayState query parameter.

URL-encode the RelayState value.

Example

```
http://www.spdemo.com:81/affwebservices/public/saml2authnrequest?
ProviderID=idp.demo&RelayState=http%3A%2F%2Fwww.spdemo.com%2Fapps%2Fapp.jsp
```

IsPassive

Determines whether the Identity Provider can interact with a user. If this query parameter is set to true, the Identity Provider must not interact with the user. Additionally, the IsPassive parameter is included with the AuthnRequest sent to the Identity Provider. If this query parameter is set to false, the Identity Provider can interact with the user.

Example

```
http://www.spdemo.com:81/affwebservices/public/saml2authnrequest?
ProviderID=idp.demo&RelayState=http%3A%2F%2Fwww.spdemo.com%
2Fapps%2Fapp.jsp&IsPassive=true
```

AssertionConsumerServiceIndex

Specifies the index of the endpoint acting as the assertion consumer. The index tells the Identity Provider where to send the assertion response.

If you use this parameter in the AuthnRequest, you cannot include the ProtocolBinding parameter also. They are mutually exclusive.

Query Parameter Processing by a SiteMinder IdP

If a Service Provider initiates single sign-on, that Service Provider can include a ForceAuthn or IsPassive query parameter in the AuthnRequest message. When a Service Provider includes these two query parameters in the AuthnRequest message, a CA SiteMinder® Identity Provider handles these query parameters as follows:

ForceAuthn Handling

When a Service Provider includes ForceAuthn=True in the AuthnRequest, a CA SiteMinder® Identity Provider takes the following actions:

- ForceAuthn=True in the AuthnRequest message, and a CA SiteMinder® session exists for a particular user. The CA SiteMinder® IdP rechallenges the user for credentials. If the user successfully authenticates, the IdP sends the identity information from the existing session in the assertion. the IdP discards the session that it generated for the reauthentication.

  A user can try to reauthenticate with different credentials than the original session. The CA SiteMinder® IdP compares the userDN and the user directory OID for the current and existing sessions. If the sessions are not for the same user, it returns a SAML 2.0 response. The response indicates that the authentication has failed.

- ForceAuthn=True in the AuthnRequest message and there is no CA SiteMinder® session. The CA SiteMinder® IdP challenges the user for credentials. If the user successfully authenticates, a session is established.

IsPassive Handling

When a Service Provider includes IsPassive in the AuthnRequest and the IdP cannot honor it, one of the following SAML responses is sent back to the Service Provider:

- IsPassive=True in the AuthnRequest message and there is no CA SiteMinder® session. The CA SiteMinder® Identity Provider returns a SAML response. This response includes an error message because CA SiteMinder® requires a session.
- IsPassive=True in the AuthnRequest message and there is a CA SiteMinder® session. The CA SiteMinder® Identity Provider returns the assertion.
- IsPassive and ForceAuthn are in the AuthnRequest message and both are set to True. The CA SiteMinder® Identity Provider returns an error because the request is invalid. IsPassive and ForceAuthn are mutually exclusive.

Configure Attributes for Assertions (optional)

Attributes can provide additional information about a user requesting access to a Service Provider resource. An attribute statement passes user attributes, DN attributes, or static data from the Identity Provider to the Service Provider in a SAML assertion. Any configured attributes are included in the assertion in one <AttributeStatement> element or the <EncryptedAttribute> element in the assertion.

Note: Attribute statements are not required in an assertion.

Servlets, web applications, or other custom applications use attributes to display customized content or enable other custom features. When used with web applications, attributes can limit the actions of a user at the Service Provider. For example, you can send an attribute variable named Authorized Amount set to a maximum dollar amount. The amount is the limit that the user can spend at the Service Provider.

Note: If CA SiteMinder® acts as a SAML 2.0 Attribute Authority as part of the assertion query/request profile, attributes are part of the authorization process. The topic Use an Attribute Authority to Authorize Users (see page 325) describes this implementation.

Attributes take the form of name/value pairs. When the Service Provider receives the assertion, it makes the attribute values available to applications.

Attributes can be made available as HTTP Headers or HTTP Cookies.

The HTTP headers and HTTP cookies have size restrictions that assertion attributes cannot exceed. The size restrictions are as follows:

- For HTTP headers, CA SiteMinder® can send an attribute in a header up to the web server size limit for a header. Only one assertion attribute per header is allowed. See the documentation for your web server to determine the header size limit.
- For HTTP cookies, CA SiteMinder® can send a cookie up to the size limit for a cookie. Each assertion attribute is sent as its own cookie. The cookie size limit is browser-specific, and that limit is for all attributes being passed to the application, not for each attribute. See the documentation for your web browser to determine the cookie size limit.

Specify Attributes for SSO Assertions

Attributes can provide information about a user requesting access to a Service Provider resource. An attribute statement passes user attributes, DN attributes, or static data from the Identity Provider to the Service Provider in a SAML assertion.

To configure an attribute

1. Navigate to the Attributes settings for the entity you are editing.
2. Click Add.

   The Add Attributes page opens.
3. From the Attribute Type drop-down list, select the name format type. This entry must match the <NameFormat> attribute in the <Attribute> element of an assertion attribute statement. The type classifies the attribute name so that the Service Provider can interpret the name.

   The options are:

   unspecified

   Determines how the name interpretation is left to your implementation.

   basic

   Indicates that the name format must use acceptable values. The acceptable values are from the values belonging to the primitive type xs:Name.

   URI

   Indicates that the name format must follow the standards for a URI reference. How the URI is interpreted is specific to the application using the attribute value.

4. From the Attribute Setup section, select one of the following options in the Attribute Kind section:
   - Static
   - User Attribute
   - DN Attribute

   The Attribute Kind selection determines the available fields in the Attribute Fields section.
5. Configure the Attribute Fields section of the page. The settings vary depending on the Attribute Kind selection. The options are:
   - Variable Name
   - Variable Value
   - Attribute Name
   - DN Spec
6. (Optional) If the attribute is retrieved from an LDAP user directory with nested groups, the Policy Server can retrieve DN attributes from the nested groups. To use nested groups, select the Allow Nested Groups check box in the Attribute Kind section.
7. (Optional) To encrypt attribute values, select the Encrypted check box.
8. For the Retrieval Method, accept the default value, SSO, to confirm that the attribute is only for single sign-on assertions.
9. Click OK to save the changes.

Use a Script to Create a New Attribute

The Advanced section of the Attribute dialog contains the Script field. This field displays the script that CA SiteMinder® generates based on your entries in the Attribute Setup section. You can copy the contents of this field and paste them into the Script field for another response attribute.

Note: If you copy and paste the contents of the Script field for another attribute, select the appropriate option in the Attribute Kind section.

Specify the Maximum Length of Assertion Attributes

The maximum length for user assertion attributes is configurable. To modify the maximum length of assertion attributes, change the settings in the EntitlementGenerator.properties file.

The property name in the file is specific to the protocol you are configuring.

Follow these steps:

1. On the system where the Policy Server is installed, navigate to *policy_server_home*\config\properties\EntitlementGenerator.properties.
2. Open the file in a text editor.
3. Adjust the maximum user attribute length for the protocols in use in your environment. The settings for each protocol are as follows:

   WS-Federation

   Property Name:
   com.netegrity.assertiongenerator.wsfed.MaxUserAttributeLength

   Property Type: Positive Integer value

   Default Value: 1024

   Description: Indicates the maximum attribute length for WS-FED assertion attributes.

   SAML 1.x

   Property Name:
   com.netegrity.assertiongenerator.saml1.MaxUserAttributeLength

   Property Type: Positive Integer value

   Default Value: 1024

   Description: Indicates the maximum attribute length for SAML1.1 assertion attributes.

   SAML 2.0

   Property Name:
   com.netegrity.assertiongenerator.saml2.MaxUserAttributeLength

   Property Type: Positive Integer value

   Default Value: 1024

   Description: Indicates the maximum attribute length for SAML2.0 assertion attributes

4. Restart the Policy Server after any change to these parameters.

Attributes for SSO and Attribute Query Requests

Indicate whether an attribute that you configure is for a single sign-on request, or for an attribute query request. The retrieval method that you configure determines the function of the attribute.

To use the same attribute for both services, create two attribute statements that use the same attribute name and variable. However, one attribute uses SSO as the retrieval method and one uses Attribute Service as the retrieval method.

Configure Single Logout (optional)

The single logout protocol (SLO) results in the simultaneous end of all sessions for a particular user, helping ensure security. These sessions must be part of the browser session that initiated the logout.

Single logout does not necessarily end all sessions for a user. For example, if the user has two browsers open, that user can establish two independent sessions. Only the session for the browser that initiates the single logout is terminated at all federated sites for that session. The session in the other browser is still active. A user-initiated logout triggers single logout.

Note: CA SiteMinder® only supports the HTTP-Redirect binding for the single logout protocol.

Configuring SLO tells the Identity Provider whether the Service Provider supports the single logout protocol, and if so, how the Service Provider handles single logout.

If you enable single logout, you must also:

- Enable the session store at the Identity Provider using the Policy Server Management Console.

  For information about the session store, see the *Policy Server Administration Guide*.
- Configure persistent sessions for the realm containing the protected resources at the Service Provider. Configure persistent sessions in the Administrative UI.

  For information about realms, see the *Policy Server Configuration Guide*.

To configure single logout

1. Log on to the Administrative UI.
2. Navigate to the SAML Profiles page for the SAML Service Provider you want to configure.
3. In the SLO section of the page, select the HTTP-Redirect check box. This setting enables single logout.

4. Enter values for the remaining fields, noting the following fields:

   Validity Duration

   Specifies the number of seconds that a single logout request is valid. This property is different from the Validity Duration for single sign-on, which is for assertions. If the validity duration expires, the IdP sends a single logout response to the entity that initiated the logout. The validity duration also depends on the skew time (set in the General tab) to calculate single logout message duration.

   SLO Location URL, SLO Response Location URL, and SLO Confirm URL

   Entries for these fields must start with https:// or http://.

5. (Optional) Select the Reuse Session Index field to use the same session index for assertions that are sent to the same partner during one browser session. This option helps ensure that single logout is successful with all third-party partners.

Federation Web Services redirects the user to the logout confirm page after the user session is removed at the Identity Provider and all Service Provider sites.

More Information:

Single Logout Request Validity (see page 223)
Guidelines for the Single Logout Confirmation Page (see page 224)

Single Logout Request Validity

The SLO validity duration and Skew Time instruct the Policy Server how to calculate the total time that the single logout request is valid.

Note: The SLO Validity Duration is a different value from the SSO Validity Duration.

The two values that are relevant in calculating the logout request duration are referred to as the IssueInstant value and the NotOnOrAfter value. In the SLO response, the single logout request is valid until the NotOnOrAfter value.

When a single logout request is generated, the Policy Server takes its system time. The resulting time becomes the IssueInstant value, which is set in the request message.

The Policy Server determines when the logout request is no longer invalid. The Policy Server takes its current system time and adds the Skew Time plus the SLO Validity Duration. The resulting time becomes the NotOnOrAfter value. Times are relative to GMT.

For example, a log out request is generated at the Identity Provider at 1:00 GMT. The Skew Time is 30 seconds and the SLO Validity Duration is 60 seconds. Therefore, the request is valid between 1:00 GMT and 1:01:30 GMT. The IssueInstant value is 1:00 GMT and the single logout request message is no longer valid 90 seconds afterward.

Guidelines for the Single Logout Confirmation Page

To support single logout, have a logout confirmation page at your site. This page lets the user know they are logged out.

The logout confirmation page must satisfy the following criteria:

- If the single logout is initiated at the Service Provider, the logout confirmation page must be an unprotected local resource at the Service Provider site.
- If single logout is initiated at an Identity Provider site, the logout confirmation page must be an unprotected local resource at the Identity Provider site.
- The page cannot be a resource in a federation partner domain. For example, if the local domain is ca.com, the SLO confirmation page cannot be in the example.com domain.

To receive feedback about a logout failure, the logout confirmation page must also support the following requirements:

- Be able to handle Base 64-encoded data and read cookies.
- Contain code that looks for a SIGNOUTFAILURE cookie. This condition must be met by the logout page at the IdP and the SP. If single logout fails, the cookie is set in the browser. The cookie contains the Partner IDs of the federation sites where logout failed. These IDs are base 64-encoded. If multiple IDs are listed, they are separated by a space character.

  By configuring the logout confirmation page to look for this cookie, the page can inform the user where the logout failed. This information is useful in networks where a user is logging out from multiple partner sites.

  Additionally, if the SIGNOUTFAILURE cookie is found, the logout confirmation page must inform users to close the web browser to remove all session data.

Configure Identity Provider Discovery at the IdP

The Identity Provider Discovery (IPD) profile provides a common discovery service that enables a Service Provider to select a unique IdP for authentication. A prior business agreement between partners is established so that all sites in the network interact with the Identity Provider Discovery service.

This profile is useful in federated networks that have more than one partner providing assertions. A Service Provider can determine which Identity Provider it sends authentication requests for a particular user.

The IdP Discovery profile is implemented using a cookie domain that is common to the two federated partners. A cookie in the agreed upon domain contains the list of IdPs that the user has visited.

For the IDP Discovery profile, the SP has to determine the IdP to which it sends authentication requests. The user that the SP wants to authenticate must have previously visited the Identity Provider and authenticated.

At the IdP, you only enable the Identity Provider Discovery feature. No other configuration is required. Enabling the feature results in a cookie being set in the common domain at the IDP Discovery Service. This process is transparent to the user.

Enable Identity Provider Discovery Profile (optional)

Federated networks can have more than one Identity Provider generating assertions. The Identity Provider Discovery profile enables users to select a specific Identity Provider for authentication.

To enable the Identity Provider Discovery Profile

1. Log on to the Administrative UI.
2. Navigate to the SAML Profiles page for the SP you want to modify.
3. In the IPD section, select the Enable checkbox.
4. Fill in the necessary fields and select the necessary settings.

   Note: Set the Service URL field to the Identity Provider Discovery Profile servlet, which is:

   https://*host:port*/affwebservices/public/saml2ipd
5. Click Submit to save your changes.

Securing the IdP Discovery Target Against Attacks

When the CA SiteMinder® Identity Provider Discovery Service receives a request for the common domain cookie, the request includes a query parameter named IPDTarget. This query parameter lists a URL where the Discovery Service must redirect to after it processes the request.

For an IdP, the IPDTarget is the SAML 2.0 Single Sign-on service. For an SP, the target is the requesting application that wants to use the common domain cookie.

We recommend protecting the IPDTarget query parameter against security attacks. An unauthorized user can place any URL in this query parameter. The URL can cause a redirection to a malicious site.

To protect the query parameter against an attack, configure the Agent Configuration Object setting **ValidFedTargetDomain**. The ValidFedTargetDomain parameter lists all valid domains for your federated environment.

Note: The ValidFedTargetDomain setting is similar to the ValidTargetDomain setting that the Web Agent uses, but this setting is defined specifically for federation.

The IPD Service examines the IPDTarget query parameter. The service obtains the domain of the URL that the query parameter specifies. The IPD Service compares this domain to the list of domains specified in the ValidFedTargetDomain parameter. If the URL domain matches one of the configured domains in the ValidFedTargetDomain, the IPD Service redirects the user to the designated URL.

If there is no domain match, the IPD Service denies the user request and they receive a 403 Forbidden in the browser. Additionally, errors are reported in the FWS trace log and the affwebservices log. These messages indicate that the domain of the IPDTarget is not defined as a valid federation target domain.

If you do not configure the ValidFedTargetDomain setting, the service redirects the user to the target URL without performing any validation.

Validate Signed Requests and Responses

The Policy Server can verify the following signed messages:

- SSO Authnrequests.
- Single logout requests and responses.

By default, signature processing is enabled because the SAML 2.0 specification requires it. Always enable signature processing in a production environment.

The Policy Server always signs SAML 2.0 POST responses and single logout requests; signing does not require configuration using the Administrative UI. The only setup that is required for signing is that you add the private key/certificate pair of the signing authority to the certificate data store.

Important! For debugging purposes only, you can temporarily disable all signature processing (both signing and verification of signatures). Select Disable Signature Processing in the Signature section of the Encryption & Signing settings.

To validate signatures of AuthnRequests from a Service Provider, or single logout requests and responses, complete the configuration steps in the Administrative UI.

To set up validation:

1. Add the public key to the certificate data store at the Identity Provider.

   The public key must correspond to the private key and certificate that the Service Provider used to do the signing.

2. In the Administrative UI, select one or both of the following check boxes:

   - Require signed AuthnRequests (Encryption & Signing settings)

     If you select this check box, the Identity Provider requires a signed authnrequest and then the IdP validates the signature of the request. If the authnrequest is not signed, the Identity Provider rejects it.

     Important: If you sign AuthnRequests, no unsolicited responses can be sent from the Identity Provider.

   - HTTP-Redirect (SAML Profiles settings)

     If you select this check box, the Identity Provider validates the signature of the SLO request and response.

3. Complete the Issuer DN and Serial Number fields (Encryption & Signing settings).

   The field values must match the certificate in the certificate data store. The certificate is the one that corresponds to the private key/certificate pair of the authority that signs the requests. To verify that you enter a matching value, view the DN of the certificate.

Encrypt a NameID and an Assertion

You can encrypt the Name ID in an assertion or the assertion itself. Encryption adds another level of protection when transmitting the assertion.

When you configure encryption, specify the partner certificate. The certificate is in the assertion. When the assertion arrives at the Service Provider, the Service Provider decrypts the encrypted data using the associated private key.

Note: If you enable encryption, the first federation call can cause the Policy Server memory to increase to load the encryption libraries and allocate additional memory.

Enabling Encryption

To implement encryption

1. Log in to the Administrative UI.
2. Navigate to the Encryption & Signing settings for the Service Provider you want to configure.
3. Configure the settings for assertion encryption.

   Be aware of the following conditions:

   - If you select rsa-oaep as an Encryption Key Algorithm, the minimum required key size is a 1024 bits.
   - To use the aes-256 bit encryption block algorithm, install Sun Java Cryptography Extension (JCE) Unlimited Strength Jurisdiction Policy Files. You can download these files from http://java.sun.com/javase/downloads/index.jsp
   - For the IssuerDN and the Serial Number fields, the IssuerDN is the DN of the certificate issuer and its associated serial number. This information locates the certificate of the Service Provider in the certificate data store. The Service Provider supplies this data.

     The IssuerDN and Serial Number that you enter must match an IssuerDN and serial number of a key/certificate pair stored in the certificate data store of the Identity Provider.
4. Click Submit to save your changes.

Request Processing with a Proxy Server at the IdP

Before the Policy Server as an IdP processes a request, it validates the message attributes using the local URL for the Federation Web Services application.

For example, an AuthnRequest message from an SP can contain the following attribute:

```
Destination="http://idp.domain.com:8080/affwebservices/public/saml2sso"
```

In this example, the destination attribute in the AuthnRequest and the address of the Federation Web Services application are the same. The Policy Server verifies that the destination attribute matches the local URL of the FWS application.

If the Policy Server sits behind a proxy server, the local and destination attribute URLs are not the same. The Destination attribute is the URL of the proxy server. For example, the AuthnRequest can include the following Destination attribute:

```
Destination="http://proxy.domain.com:9090/affwebservices/public/saml2sso"
```

The local URL for Federation Web Services, http://idp.domain.com:8080/affwebservices/public/saml2sso, does not match the Destination attribute so the Policy Server denies the request.

You can specify a proxy configuration to alter how the Policy Server determines the local URL for verifying a message attribute. If you specify a proxy, the system replaces the *protocol*://*authority* portion of the local URL with the proxy server URL. The result is a match between the two URLs.

Configure Request Processing with a Proxy Server

The Policy Server can sit behind a proxy server. For this deployment, configure the proxy so that the system finds a match between the URL in a request message attribute and the local proxy URL. There must be a match to process the request. The Policy Server replaces the *protocol*://*authority* portion of the local URL with the proxy server URL, which results in a match between the two URLs.

To use a proxy server at the IdP

1. Log in to the Administrative UI.
2. Navigate to the General settings for the Service Provider you want to configure.

3. Enter a partial URL for the proxy server, of the form *protocol*://*authority*.

   For example, the proxy server configuration would be:

   ```
   http://proxy.domain.com:9090
   ```

   If your network includes the SPS federation gateway, the Server field must specify the SPS federation gateway host and port, for example,

   ```
   http://sps_gateway_server.ca.com:9090
   ```

4. Click Submit to save your changes.

The value that you enter for the Server field affects the URLs for the following IdP services:

- Single Sign-On Service
- Single Logout Service
- Artifact Resolution Service
- Attribute Service
- Authentication URL—use the proxy server URL. After the Policy Server authenticates the user, it redirects the user to the proxy server to get to the Single Sign-On service.

The Server value becomes part of the URL used to verify SAML attributes like the Destination attribute. If you are using a proxy server for one URL, use it for all these URLs.

Chapter 15: Configure a SAML 2.0 Service Provider

This section contains the following topics:

Service Provider Setup

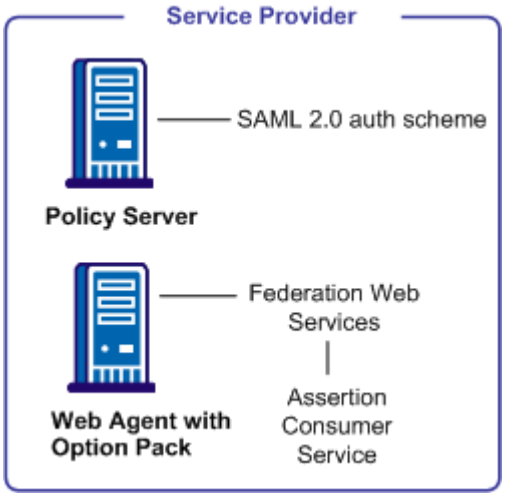
CA SiteMinder® or the SPS federation gateway can act as a SAML 2.0 Service Provider. The Service Provider uses the assertions that it receives from an Identity Provider to authenticate users and then provide access to the requested federation resources. Assuming that the CA SiteMinder® Service Provider has access to a user store at its site, the Service Provider uses the CA SiteMinder® SAML 2.0 authentication scheme to authenticate users.

The SAML 2.0 authentication scheme enables cross-domain single sign-on. The Service Provider is able to consume an assertion from an Identity Provider, identify a user, and establish a CA SiteMinder® session. After a session is established, the Service Provider can authorize the user for specific resources.

The following illustration shows the components for authentication at the Service Provider.

Note: A site can be both an Identity Provider and a Service Provider.

The major components for SAML 2.0 authentication are shown in the following illustration.



Note: The SPS federation gateway can replace the Web Agent and Web Agent Option Pack to provide the Federation Web Services application functions. For information about installing and configuring the SPS federation gateway, see the *Secure Proxy Server Administration Guide*.

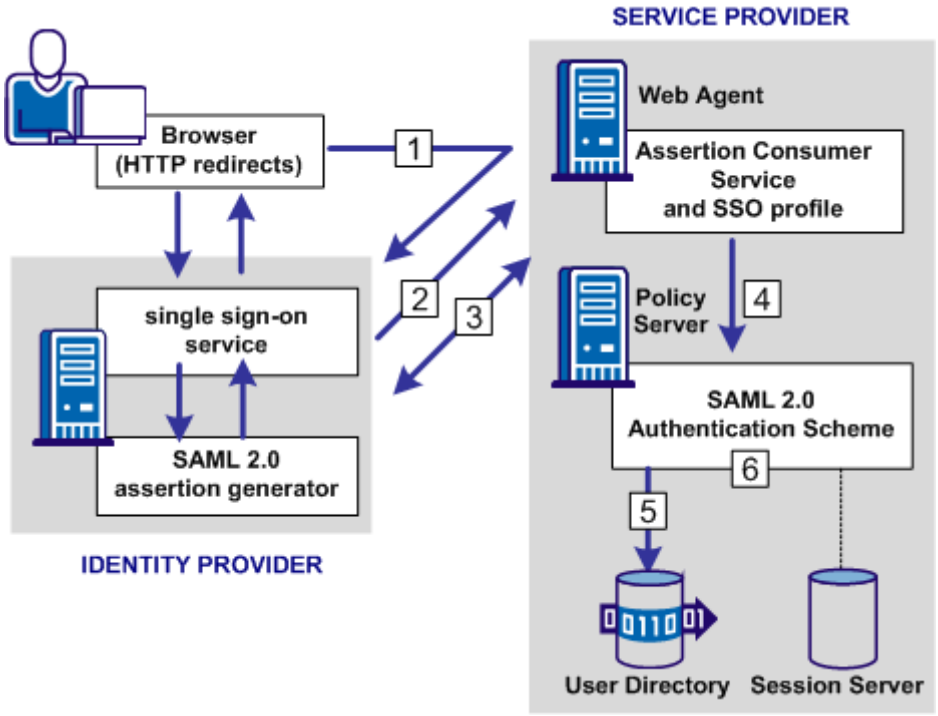
The SAML 2.0 authentication scheme is configured at the Policy Server that resides at the Service Provider site. The authentication scheme invokes the Assertion Consumer Service, a component of the Federation Web Services application, that is installed on the Web Agent or SPS federation gateway at the Service Provider site. The service obtains information from the SAML authentication scheme, then uses that information to extract the necessary information from a SAML assertion.

The SAML assertion becomes the user credentials to log in to the Service Provider Policy Server. The user is authenticated and authorized, and if authorization is successful, the user is redirected to the target resource.

The Assertion Consumer Service accepts AuthnRequests that include an AssertionConsumerServiceIndex value of 0. All other values for this setting are denied.

SAML Authentication Request Process

The following illustration shows how the SAML 2.0 authentication scheme processes requests.



Note: The SPS federation gateway can replace the Web Agent and Web Agent Option Pack to provide the Federation Web Services application functions. For information about installing and configuring the SPS federation gateway, see the *Secure Proxy Server Administration Guide*.

The functional flow for authentication is as follows:

1. A user makes a request for a Service Provider resource. This request goes to the AuthnRequest service at the Service Provider. The request is then redirected to the Identity Provider to obtain a SAML assertion.
2. The Identity Provider returns a response to the Service Provider.

   For HTTP-POST binding, the response contains the assertion. For the HTTP-Artifact binding, the response contains a SAML artifact.

3. The Assertion Consumer Service at the Service Provider receives the response message and determines whether the POST or Artifact binding is being used.

   For the HTTP-Artifact binding, the Assertion Consumer Service sends the artifact to the Identity Provider to retrieve the assertion. The Identity Provider returns a response that contains the assertion. The Assertion Consumer Service uses the response with the assertion as credentials to the Policy Server.

4. The Policy Server invokes the SAML 2.0 authentication scheme by passing the response message with the user credentials to the scheme to be authenticated.

5. The user disambiguation process begins.

6. After the disambiguation phase is complete, the SAML 2.0 authentication scheme validates the credentials in the assertion. The scheme also validates the assertion for time validity, and, if applicable, verifies that a trusted Identity Provider signed the assertion.

   Note: For the POST binding, a signature is required. If a signature is not present, authentication fails. For the Artifact binding, a signed assertion is optional because the assertion is obtained over a secure channel between the Service Provider and Identity Provider.

   If single logout is enabled, the SLO servlet redirects the user to a No Access URL.

Prerequisites for a Relying Partner

For CA SiteMinder® to act as the relying partner, complete following tasks:

- Install the Policy Server.
- Install one of the following components:
  - The Web Agent and the Web Agent Option Pack. The Web Agent authenticates users and establishes a session. The Option Pack provides the Federation Web Services application. Be sure to deploy the FWS application on the appropriate system in your network.
  - The SPS federation gateway, which has an embedded Web Agent and has the Federation Web Services application on the embedded Tomcat web server.

  For more information, see the *Web Agent Option Pack Guid*e.
- Private keys and certificates are imported for functions that require verification and encrypting of messages.
- An asserting partner is set up within the federated network.

How to Configure a SAML 2.0 Authentication Scheme

Configuring a Service Provider requires the following tasks:

1. Complete the SAML 2.0 authentication scheme prerequisites.
2. Select the authentication scheme type (see page 236).
3. Configure disambiguation to authenticate users.
4. Configure single sign-on (see page 240).

Configure a SAML authentication scheme for each Identity Provider that is a federation partner and generates assertions. Bind each scheme to a realm. The realm consists of all the URLs of the target resources requested by users. Protect these resources with a policy.

Tips:

- Certain parameter values at the Identity Provider and Service Provider must match for the configuration to work. A list of those parameters is available in Configuration Settings that Must Use the Same Values (see page 361).
- Verify that you are using the correct URLs for the Federation Web Services servlets. The URLs are listed in Federation Web Services URLs Used in SiteMinder Configuration (see page 367).

Optional Configuration Tasks for a Service Provider

The optional tasks for configuring CA SiteMinder® as a Service Provider are:

- Enable single logout (see page 245).
- Enable encryption for Name IDs and/or assertions (see page 247).
- Sign artifact resolve message (see page 247).
- Require signed artifact response (see page 247).
- Customize assertions using the Message Consumer Plug-in (see page 157).

Navigating Legacy Federation Dialogs

The Administrative UI provides two ways to navigate to the legacy federation configuration dialogs.

You can navigate in one of two ways:

- Following a wizard to configure a new legacy federation object.

  When you create an object, a page displays with a configuration wizard. Follow the steps in the configuration wizard to create the object.

- Selecting tabs to modify an existing legacy federation object.

  When you modify an existing object, a page displays with a series of tabs. Modify the configuration from these tabs. These tabs are the same as the steps in the configuration wizard.

Select the Authentication Scheme Type

The Service Provider uses the identity information in the assertion to authorize access to protected federated resources. A SAML authentication scheme is used for this process.

Before you can assign a SAML 2.0 authentication scheme to protect resources, configure the scheme.

Follow these steps:

1. Review the SAML 2.0 Authentication Scheme Prerequisites.
2. Log in to the Administrative UI.
3. Navigate to Infrastructure, Authentication, Authentication Schemes.

   The Authentication Scheme page opens at the General settings.
4. Name the authentication scheme.
5. In the Authentication Scheme Type drop-down list, select SAML 2.0 Template. You can also select a protection level for this scheme.

   The contents of the Authentication Scheme dialog change to support the SAML 2.0 scheme.
6. In the Scheme Setup section, click SAML 2.0 Configuration to define the details of the authentication scheme.

   If you are configuring the scheme for the fist time, follow the configuration wizard to set up the authentication scheme.

Specify the General Information for the SAML 2.0 Auth Scheme

Identity the Service Provider and Identity Provider in the General settings for the SAML 2.0 authentication schemes.

Follow these steps:

1. From the main authentication scheme page, click SAML 2.0 Configuration.

   If you are modifying an existing scheme, click Modify then click SAML 2.0 Configuration.

   The detailed settings for the scheme display.
2. In the General settings, complete the required fields.
3. Move on to the User Disambiguation section.

Locate User Records for SAML 2.0 Authentication

When you configure an authentication scheme, you define a way for the authentication scheme to look up a user in the local user store. After the correct user is located, the system generates a session for that user. Locating the user in the user store is the process of disambiguation. How the Policy Server disambiguates a user depends on the configuration of the authentication scheme.

For successful disambiguation, the authentication scheme first determines a LoginID from the assertion. The LoginID is a CA SiteMinder®-specific term that identifies the user. By default, the LoginID is extracted from the Name ID in the assertion. You can also obtain the LoginID using an Xpath query.

After the authentication scheme determines the LoginID, the Policy Server checks if a search specification is configured for the authentication scheme. If no search specification is defined for the authentication scheme, the LoginID is passed to the Policy Server. The Policy Server uses the LoginID together with the user store search specification to locate the user. For example, the LoginID value is Username and the LDAP search specification is set to the uid attribute. The Policy Server uses the uid value (Username=uid) to locate the user.

If you configure a search specification for the authentication scheme, the LoginID is not passed to the Policy Server. Instead, the search specification is used to locate a user.

You can configure user disambiguation in one of two ways:

- Locally, as part of the authentication scheme
- By selecting a configured SAML affiliation

Configure Disambiguation Locally as Part of the Authentication Scheme

If you choose to disambiguate locally, there are two steps in the process:

1. Obtain the LoginID by the default behavior or by using an Xpath query.
2. Locate the user in the user store by the default behavior or defining a user lookup.

Note: The use of Xpath and a search specification are optional.

Obtain the LoginID

You can find the LoginID in two ways:

- Relying on the default behavior, where the LoginID is extracted from the NameID in the assertion. This option requires no configuration.
- Using an Xpath query to find the LoginID in place of the default behavior.

To use an Xpath query to determine the LoginID

1. Navigate to the SAML 2.0 authentication scheme.
2. Click SAML 2.0 Configuration.
3. From the SAML 2.0 properties page, enter an Xpath query that the authentication scheme uses to obtain a LoginID. Click OK to save your changes.

   Xpath queries must not contain namespace prefixes. The following example is an invalid Xpath query:

   ```
   /saml:Response/saml:Assertion/saml:AuthenticationStatement/
   saml:Subject/saml:NameIdentifier/text()
   ```

   The valid Xpath query is:

   ```
   //Response/Assertion/AuthenticationStatement/Subject/
   NameIdentifier/text()
   ```

Configure a User Lookup to Locate a User

After you obtain the LoginID, you can configure a user lookup to locate the user in place of the default behavior, where the LoginID is passed to the Policy Server.

To locate a user with a search specification

1. Navigate to the SAML 2.0 authentication scheme.
2. Click SAML 2.0 Configuration.

3. In the User Lookup section, enter a search specification in the appropriate namespace field. The search specification defines the attribute that the authentication scheme uses to search a namespace. Use %s as the entry representing the LoginID.

   For example, the LoginID has a value of user1. If you specify Username=%s in the Search Specification field, the resulting string is Username=user1. This string is verified against the user store to find the correct record for authentication.

4. Click OK to save your configuration changes.

Use a SAML Affiliation to Locate a User Record (Optional)

A group of Service Providers can form an affiliation. Grouping Service Providers establishes an association across the federated network, such that a relationship with one member of an affiliation establishes a relationship with all members of the affiliation.

All Service Providers in an affiliation share the name identifier for a single principal. If one Identity Provider authenticates a user and assigns that user an ID, all members of the affiliation use that same name ID. The single name ID reduces the configuration that is required at each Service Provider. Additionally, using one name ID for a principal saves storage space at the Identity Provider.

You can use the optional Xpath query and search specification for user disambiguation. These options are defined as part of the affiliation itself and not part of the authentication scheme.

Note: Define an affiliation first before using it in an authentication scheme configuration.

To select an affiliation

1. Navigate to the SAML 2.0 authentication scheme page.
2. Click SAML 2.0 Configuration.
3. In the General settings.
4. In the User Disambiguation section, select a predefined affiliation in the SAML Affiliation drop-down field. These affiliations are configured at the Identity Provider.

Configure Single Sign-on at the SP

To establish single sign-on between the Identity Provider and the Service Provider, specify the SSO bindings.

The SSO settings let you configure single sign-on using the artifact or POST binding.

Part of the single sign-on configuration is defining the Redirect Mode setting. The Redirect Mode specifies how CA SiteMinder® sends assertion attributes, if available, to the target application. You can send assertion attributes as HTTP Headers or HTTP cookies.

The HTTP headers and HTTP cookies have size restrictions that assertion attributes cannot exceed. The size restrictions are as follows:

- For HTTP headers, CA SiteMinder® can send an attribute in a header up to the web server size limit for a header. Only one assertion attribute per header is allowed. See the documentation for your web server to determine the header size limit.
- For HTTP cookies, CA SiteMinder® can send a cookie up to the size limit for a cookie. Each assertion attribute is sent as its own cookie. The cookie size limit is browser-specific, and that limit is for all attributes being passed to the application, not for each attribute. See the documentation for your web browser to determine the cookie size limit.

To configure single sign-on

1. Navigate to the SAML 2.0 authentication scheme.
2. Click SAML 2.0 Configuration, SSO.
3. Complete entries for the SSO fields.
4. (Optional) Specify a target resource for single sign-on to work. The target specifies the requested resource at the destination Service Provider site.

   The Service Provider does not have to use the default target. The link that initiates single sign-on can contain a query parameter that specifies the target.

5. In the Bindings section, you can select HTTP-Artifact and HTTP-Post.

   If HTTP-POST is selected and artifact is not selected, only the POST binding is accepted from the Identity Provider. If no binding is specified, the default is HTTP-artifact.

   If you select HTTP-Artifact binding:

   - Enable the session server (see page 97) to store the assertion before it is retrieved.
   - Configure the back channel (see page 243). Select the type of authentication scheme protecting communication with the Artifact Resolution Service. This service retrieves the assertion at the Identity Provider.
   - Optionally, specify an integer as an Index entry for each binding.

     If you have multiple endpoints, you can configure indexed endpoints. The Service Provider includes the specified endpoint entry as a query parameter in the AuthnRequest. The AuthnRequest is sent to the single sign-on service at the Identity Provider.

More Information:

Configure the Authentication Scheme that Protects the Artifact Service (see page 133)

Enforcing a Single Use Policy to Enhance Security

A single use policy prevents SAML 2.0 assertions from being reused at a Service Provider to establish a second session. This feature applies to assertions that arrive by way of the POST binding.

Note: Single use policy feature is enabled by default when you select the HTTP-POST binding.

Designating an assertion for one time use is an additional security measure for authenticating across a single sign-on environment. From a browser, an attacker can acquire a SAML assertion that has been used to establish a CA SiteMinder® session. The attacker can then POST the assertion to the Assertion Consumer Service at the Service Provider to establish a second session. However, if the assertion is designated for one-time use, this type of risk is mitigated.

CA SiteMinder® enforces a single use policy using expiry data. Expiry data is time-based data about the assertion. The SAML 2.0 authentication scheme stores the expiry data in the session store. Expiry data verifies that a SAML 2.0 POST assertion is only used a single time.

How the Single Use Policy is Enforced

Upon successful validation of a SAML 2.0 assertion, the authentication scheme writes assertion data in the expiry data table. The data includes an assertion ID key and an expiration time. The session store management thread in the Policy Server deletes expired data from the expiry data table.

If the scheme tries to validate assertion data and an expiry data entry has the same assertion ID key, writing assertion data fails. If the scheme cannot write to the expiry table, the SAML 2.0 authentication scheme denies the authentication in the same manner as an invalid assertion.

If the database is unavailable, single use of the assertion cannot be enforced. Consequently, the authentication scheme denies the request and the assertion is not reused.

Configure a Single Use Policy

To configure a single use policy

1. Navigate to the SAML 2.0 authentication scheme.

   Click Modify, SAML 2.0 Configuration.

2. Select to the SSO tab.

3. In the HTTP-Post, section the Enforce Single Use Policy check box is selected by default.

4. Enable the session store.

   For instructions on enabling the session store, see the *Policy Server Administration Guide*.

More Information:

Storing User Session, Assertion, and Expiry Data (see page 97)
Enforcing a Single Use Policy to Enhance Security (see page 241)

Permit the Creation of a Name Identifier for SSO

As part of a single sign-on request, a Service Provider can generate an AuthnRequest that includes an attribute named AllowCreate, which is set to true. The Service Provider wants to obtain an identity for the user. Upon receiving the AuthnRequest, the Identity Provider generates an assertion. The Identity Provider searches the appropriate user record for the assertion attribute serving as the Name ID. If the Identity Provider cannot find a value for the NameID attribute, it generates a persistent identifier. The Allow/Create feature enables the creation of the identifier.

The persistent identifier is a randomly generated ID. The Identity Provider uses this identifier as the value of the Name ID attribute and places it in the assertion. The Identity Provider then returns the assertion to the Service Provider. For example, if the NameID attribute is set to telephone and there is no value for telephone in the user record, the NameID is set to the randomly generated identifier.

When the Service Provider receives the assertion, the SAML 2.0 authentication scheme processes the response. The scheme then performs a user lookup in its local user store. If the Service Provider locates the user record, it grants the user access.

Enable the Allow/Create feature at the Identity Provider for the Identity Provider to generate a unique identifier. The Identity Provider only generates the identifier if the feature is enabled. The normal flow of assertion generation continues after an entry is made in the Identity Provider log file that a unique identifier was not created.

Include an Allow/Create Attribute in Authentication Requests

To permit the Identity Provider to create an identifier for the Name ID, include the Allow/Create attribute in the AuthnRequest message.

Note: The administrator at the Identity Provider must enable the Allow/Create feature for the identifier to be generated.

Follow these steps:

1. Navigate to the SAML 2.0 authentication scheme.
2. Click SAML 2.0 Configuration, SSO.
3. Check the Allow IDP to Create New Identifier check box.
4. Click OK.

Configure the Back Channel for HTTP-Artifact SSO

If you select the HTTP-Artifact binding for single sign-on, select an authentication scheme to secure the back channel to the Artifact Resolution Service. This service retrieves the assertion at the Identity Provider.

To configure the back channel

1. Navigate to the SAML 2.0 authentication scheme.
2. Click SAML 2.0 Configuration, Encryption and Signing.

3. In the Backchannel section, complete all the fields.

   Important! If you are using basic authentication for the backchannel authentication scheme, the value of the SP Name field is the name of the Service Provider. No additional configuration is necessary. If you are using client certificate authentication, the SP Name field must be the alias of the client certificate stored in the certificate data store. The SP uses the certificate as a credential to gain access to the Artifact Resolution Service.

4. Click OK to save your configuration.

More Information:

Enable Client Certificate Authentication for the Back Channel (optional) (see page 263)

ECP Configuration at the Service Provider

To configure ECP, enable the feature at the Identity Provider and the Service Provider. The following procedure is for a CA SiteMinder® Service Provider.

For more information about ECP, read the overview (see page 198).

Follow these steps:

1. Direct requests for a protected resource to the AuthnRequest service at the Service Provider. The following URL shows an example:

   https://*host*:*port*/affwebservices/public/saml2authnrequest

2. Log in to the Administrative UI at the Service Provider.
3. Modify the relevant SAML 2.0 authentication scheme object.
4. Click SAML 2.0 Configuration in the Scheme Setup section.

   The configuration tabs for the scheme are displayed.

5. Select the SSO tab.
6. Select the Enhanced Client and Proxy Profile check box then click OK.
7. Click Submit to save the changes.

The CA SiteMinder® Service Provider can now process ECP calls.

Note: A single SAML Service Provider object can handle artifact, POST, SOAP, and PAOS bindings for single sign-on requests. SOAP and PAOS are the bindings for the ECP profile. The Identity Provider and Service Provider determine the binding being used based on the parameters in a request.

Enable Single Logout

The single logout (SLO) profile allows near-simultaneous logout of all sessions that a specific session authority provides and which are associated with a particular user. The user initiates the logout directly. A session authority is the authenticating entity that has initially authenticated the user. In most cases, the session authority is the Identity Provider.

Single logout helps ensure that no sessions are left open for unauthorized users to gain access to resources at the Service Provider.

The user can initiate single logout service from a browser by clicking a link at the Service Provider or at the Identity Provider. The user clicks the logout link which points to an SLO servlet. This servlet, which is a component of Federation Web Services, processes logout requests and responses coming from a Service Provider or Identity Provider. The servlet does not need to know the originator of the request or response. The servlet uses the CA SiteMinder® session cookie to determine the session to log out.

Bindings for Single Logout

The single logout feature transports messages using the HTTP-Redirect binding. This binding determines how SAML protocol messages are transported using HTTP redirect messages, which are 302 status code responses.

Configure Single Logout

If you enable single logout at the Service Provider, configure persistent sessions for the realm containing the protected resources at the Service Provider. Configure persistent sessions in the Administrative UI.

To configure single logout

1. Navigate to the SAML 2.0 authentication scheme.
2. Click SAML 2.0 Configuration, SLO.
3. In the SLO section of the page, select the HTTP-Redirect check box. The other single logout settings become active.
4. Enter values for the remaining fields, noting the following information:

   Validity Duration

   Specifies the number of seconds that a single logout request is valid. If the validity duration expires, a single logout response is generated. The response is sent to the entity who initiated the logout. The validity duration also depends on the skew time to calculate single logout message duration.

SLO Location URL, SLO Response Location URL, and SLO Confirm URL

Entries for these fields must start with https:// or http://.

Note: Click Help for descriptions of settings and controls, including their respective requirements and limits.

After single logout is initiated, the user session is removed at the Identity Provider and all Service Provider sites. Federation Web Services then redirects the user to the logout confirm page.

More Information:

Storing User Session, Assertion, and Expiry Data (see page 97)

Digital Signing Options at the Service Provider

The SAML 2.0 authentication scheme configuration includes digital signing options for the following transactions:

- Authentication requests
- Single logout requests and responses
- Artifact resolve messages—for resolution of a SAML artifact to retrieve the assertion.
- Attribute queries—for authorizations taking place between an Attribute Authority (IdP) and a SAML Requester(SP).

By default, signature processing is enabled because the SAML 2.0 specification requires signing. For debugging your initial federation setup *only*, you can temporarily disable all signature processing for the Service Provider (signing and verification of signatures) by selecting the Disable Signature Processing option. After debugging is complete, reenable signature processing.

Important! If you disable signature processing in a production environment, you are disabling a mandatory security function.

To specify the signing options

1. Navigate to the SAML 2.0 authentication scheme.
2. Click SAML 2.0 Configuration, Encryption & Signing.

3. Complete the fields in the D-sig Info section. Note the following information:
   - For HTTP-POST (single sign-on), enter information about the certificate that validates the signature of the posted assertion. The Issuer DN and the Serial Number together identify the certificate corresponding to the private key the IdP used to sign the assertion.

     The value that you enter for the Issuer DN field must match the issuer DN of the certificate in the certificate data store.
   - For HTTP-Redirect (single logout), enter information about the certificate that validates the signature of the SLO request.
4. Complete the settings in the Signature Processing section of the dialog.
5. For HTTP-Artifact single sign-on only, configure the back channel settings.
6. Click OK.

Enforce Assertion Encryption Requirements for Single Sign-on

The encryption feature specifies that the authentication scheme processes only an encrypted assertion or Name ID in the assertion.

For added security, the Identity Provider can encrypt the Name ID, user attributes, or the entire assertion. Encryption adds another level of protection when transmitting the assertion. When encryption is enabled at the Identity Provider, the certificate (public key) is used to encrypt the data. When the assertion arrives at the Service Provider, it decrypts the encrypted data with the associated private key.

When you configure encryption at the Session Provider, the assertion must contain an encrypted Name ID or assertion or the Service Provider rejects the assertion.

Set Up Encryption for SSO

You can enforce encryption requirements for the assertion.

To enforce encryption requirements

1. Navigate to the SAML 2.0 authentication scheme.
2. Click SAML 2.0 Configuration, Encryption & Signing.

   The encryption and signing settings page displays.
3. To require an encrypted Name ID, select the Require Encrypted Name ID check box.
4. To require an encrypted assertion, select the Require Encrypted Assertion check box.

   You can select the Name ID and the assertion.

5. (Optional) Specify an alias for the private key that decrypts any encrypted data in the assertion received from the Identity Provider.
6. Click OK to save your changes.

Without any encryption requirements, the Service Provider accepts Name IDs and assertions that are encrypted or in clear text.

Create a Custom SAML 2.0 Authentication Scheme (optional)

You can use a custom SAML 2.0 scheme that is written with the CA SiteMinder® Authentication API instead of the existing SAML 2.0 authentication template.

The main authentication scheme page includes the Library field in the Scheme Setup section of the page. This field contains the name of the shared library that processes SAML artifact authentication. Do not change this value unless you have a custom authentication scheme.

The default shared library for HTML Forms authentication is smauthhtml.

IDP Discovery Configuration at the Service Provider

The Identity Provider Discovery (IPD) profile provides a common discovery service that enables a Service Provider to select a unique IdP for authentication. A prior business agreement between partners is established so that all sites in the network interact with the Identity Provider Discovery service.

This profile is useful in federated networks that have more than one partner providing assertions. A Service Provider can determine which Identity Provider it sends authentication requests for a particular user.

The IdP Discovery profile is implemented using a cookie domain that is common to the two federated partners. A cookie in the agreed upon domain contains the list of IdPs that the user has visited.The SP has to redirect the user to the IdP Discovery Service to retrieve the common domain cookie. The cookie contains the list of IdPs that the user has already visited. From this list, the SP chooses the correct Identity Provider and then sends an AuthnRequest.

Note: The user requiring authentication must have previously visited the Identity Provider and authenticated.

IDP Discovery occurs as follows:

1. The browser requests the site selection page at the SP.

   This site selection page is aware of the IDP Discovery Service URL.

2. The site selection page redirects the user to the IDP Discovery Service URL in the common domain. The redirect URL contains a query parameter indicating that it wants the Common Domain Cookie.

3. The IDP Discovery Service retrieves the value of the Common Domain Cookie and sets it as a query parameter. The service then redirects the user back to the site selection page at the SP.

4. The SP populates the site selection page with IdP IDs, which are URIs at which the user has previously authenticated.

5. The user selects an IdP to perform the user authentication.

More information:

Configure Identity Provider Discovery at the IdP (see page 225)

Configure Identity Provider Discovery at the SP

Configuration of the Identity Provider Discovery profile at the Service Provider does not involve the Administrative UI. The profile is enabled in the Administrative UI at the Identity Provider.

The first step in the process is to create a site selection page. The Policy Server comes with a sample site selection page, named IdPDiscovery.jsp, that the SP can use.

The first link on the site selection page redirects the browser from one domain to the IdP Discovery service in the common domain. The service gets the common domain cookie, named **_saml_idp**. When the IdP Discovery Service at the SP receives the request, it gets the common domain cookie. The service adds the common domain cookie as a query parameter to the link. The service then redirects the user back to the IdPDiscovery.jsp site selection page in the regular domain.

By default, the IdPDiscovery.jsp page displays only a list of IDs for the IdPs that it extracts from the common cookie. The list is static; there are no HTML links associated with the list that initiate communication with the associated IdP.

To configure IdP Discovery at the SP

1. Create a site selection page that requests the Common Domain Cookie from the IdP Discovery Service at the SP.

   The Policy Server comes with a sample site selection page, named IdPDiscovery.jsp that the SP can use to implement IdP Discovery. You can find the page in the following directory:

   web_agent_home/affwebservices/public

2. Edit the following link on the sample page for your SP site. The first part of the link specifies the common domain where the saml2idp cookie resides. The second part of the link specifies the regular domain where the IdPDiscovery.jsp resides.

   For example:

   <a href="**http://myspsystem.commondomain.com**/affwebservices/public
   /saml2ipd/?IPDTarget=/**http://myspsystem.spdomain.com**/affwebservices
   /public/IdpDiscovery.jsp&SAMLRequest=getIPDCookie">
   Retrieve idp discovery cookie from IPD Service

   When the user is redirected back to the regular domain with the target site selection page, it now has the common cookie.

3. (Optional) Edit the IdPDiscovery.jsp site selection page so it displays an HTML link for each IdP. Each link triggers an AuthnRequest to the IdP to initiate single sign-on. By default, the IdPDiscovery.jsp page only displays a list of IDs for the IdPs that it extracts from the common cookie.

4. Use the edited site selection page to test IdP Discovery.

With IdP Discovery working, the site selection page displays a list of IdPs from which to select.

Securing the IdP Discovery Target Against Attacks

When the CA SiteMinder® Identity Provider Discovery Service receives a request for the common domain cookie, the request includes a query parameter named IPDTarget. This query parameter lists a URL where the Discovery Service must redirect to after it processes the request.

For an IdP, the IPDTarget is the SAML 2.0 Single Sign-on service. For an SP, the target is the requesting application that wants to use the common domain cookie.

We recommend protecting the IPDTarget query parameter against security attacks. An unauthorized user can place any URL in this query parameter. The URL can cause a redirection to a malicious site.

To protect the query parameter against an attack, configure the Agent Configuration Object setting **ValidFedTargetDomain**. The ValidFedTargetDomain parameter lists all valid domains for your federated environment.

Note: The ValidFedTargetDomain setting is similar to the ValidTargetDomain setting that the Web Agent uses, but this setting is defined specifically for federation.

The IPD Service examines the IPDTarget query parameter. The service obtains the domain of the URL that the query parameter specifies. The IPD Service compares this domain to the list of domains specified in the ValidFedTargetDomain parameter. If the URL domain matches one of the configured domains in the ValidFedTargetDomain, the IPD Service redirects the user to the designated URL.

If there is no domain match, the IPD Service denies the user request and they receive a 403 Forbidden in the browser. Additionally, errors are reported in the FWS trace log and the affwebservices log. These messages indicate that the domain of the IPDTarget is not defined as a valid federation target domain.

If you do not configure the ValidFedTargetDomain setting, the service redirects the user to the target URL without performing any validation.

Customize Assertion Processing with the Message Consumer Plug-in

The message consumer plug-in is a Java program that implements the Message Consumer Plug–in. The plug-in lets you implement your own business logic for processing assertions, such as rejecting an assertion and returning a status code. This additional processing works together with the standard processing of an assertion.

Note: For more information about status codes for authentication and disambiguation, see the *CA SiteMinder® Programming Guide for Java*.

During authentication, CA SiteMinder® first tries to process the assertion by mapping a user to its local user store. If CA SiteMinder® cannot find the user, it calls the postDisambiguateUser method of the message consumer plug-in.

If the plug-in successfully finds the user, CA SiteMinder® proceeds to the second phase of authentication. If the plug-in cannot map the user to a local user store, the plug-in returns a UserNotFound error. The plug-in can optionally use the redirect URL feature. Without the consumer plug-in, the redirect URLs are based on the error that the SAML authentication scheme generates.

During the second phase of authentication, CA SiteMinder® calls the postAuthenticateUser method of the message consumer plug-in, if the plug-in is configured. If the method succeeds, CA SiteMinder® redirects the user to the requested resource. If the method fails, you can configure the plug-in to send the user to a failure page. The failure page can be one of the redirect URLs that you can specify with the authentication scheme configuration.

Additional information about the message consumer plug-in can be found as follows:

- Reference information (method signatures, parameters, return values, data types), and the constructor for UserContext class, are in the *Java Developer Reference*. Refer to the MessageConsumerPlugin interface.
- Overview and conceptual information about authentication and authorization APIs, see the *CA SiteMinder® Programming Guide for Java*.

To configure the plugin

1. Install the CA SiteMinder® SDK, if you have not done so already.
2. Implement the MessageconsumerPlugin.java interface, which is part of the CA SiteMinder® SDK.
3. Deploy your message consumer plug-in implementation class.
4. Enable the message consumer plug-in in the Administrative UI.

Implement the MessageConsumerPlugin Interface

Create a custom message consumer plug-in by implementing the MessageConsumerPlugin.java interface. The minimum requirements for the implementation class are listed in the following procedure.

Follow these steps:

1. Provide a public default constructor method that contains no parameters.
2. Provide code so that the implementation is stateless. Many threads must be able to use a single plug-in class.

3. Implement methods in the interface as your requirements demand.

   The MessageConsumerPlugin includes the following four methods:

   init()

   Performs any initialization procedures that the plug-in requires. CA SiteMinder® calls this method once for each plug-in instance, when the plug-in is loaded.

   release()

   Performs any rundown procedures that the plug-in requires. CA SiteMinder® calls this method once for each plug-in instance, when CA SiteMinder® is shutting down.

   postDisambiguateUser()

   Provides processing to disambiguate a user when the authentication scheme is unable to do so. Alternatively, this method can add data for new federation users to a user store. This method receives the decrypted assertion. The decrypted assertion is added to the properties map passed to plug-in under the key "_DecryptedAssertion".

   postAuthenticateUser()

   Provides additional code to determine the final outcome of assertion processing, regardless of whether the Policy Server processing is a success or failure.

CA SiteMinder® provides the following samples of the Message Consumer plug-in class:

MessageConsumerPluginSample.java in *installation_home*\sdk\samples\messageconsumerplugin

MessageConsumerSAML20.java in *installation_home*\sdk\samples\authextensionsaml20

Deploy a Message Consumer Plug-in

After you have coded your implementation class for the MessageConsumerPlugin interface, compile it and verify that CA SiteMinder® can find your executable file.

To deploy the Message Consumer Plugin:

1. Compile the MessageConsumerPlugin Java file. The file requires the following dependent libraries, which are installed with the Policy Server:

   installation_home\siteminder\bin\jars\SmJavaApi.jar

   An identical copy of SmJavaApi.jar is installed with CA SiteMinder® SDK. The file is in the directory *installation_home*\sdk\java\SmJavaApi.jar.

   You can use either of them at development time.

2. When a plug-in class is available, in a folder or a jar file, modify the -Djava.class.path value in the JVMOptions.txt file. This step enables the plug-in class to load with the modified classpath. Locate the JVMOptions.txt file in the directory *installation_home*\siteminder\config.

   Note: Do not modify the classpath for the existing xerces.jar, xalan.jar, or SmJavaApi.jar.

3. Restart the Policy Server to pick up the latest version of MessageConsumerPlugin. This step is necessary each time the plug-in Java file is recompiled.

4. Enable the plug-in.

Enable the Message Consumer Plug-in for SAML 2.0

After writing a message consumer plug-in and compiling it, enable the plug-in by configuring settings in the Administrative UI. The UI settings tell CA SiteMinder® where to find the plug-in.

Do not configure the plug-in settings until you deploy the plug-in (see page 159).

To enable the message consumer plug-in

1. Log on to the Administrative UI
2. Navigate to the SAML 2.0 authentication scheme dialog.
3. Click Advanced.
4. In the Message Consumer Plugin section, complete the following fields:

   Full Java Class Name

   Specify the Java class name for the plug-in, For example, a sample class included with the CA SiteMinder® SDK is:

   com.ca.messageconsumerplugin.MessageConsumerPluginSample

   Parameter

   Specify a string of parameters that are passed to the plug-in specified in the Full Java Class Name field.

   As an alternative to configuring the plug-in in the Administrative UI, use the Policy Management API (C or Perl) to set the IdpPluginClass and IdpPluginParameters.

5. Restart the Policy Server.

Supply SAML Attributes as HTTP Headers

An assertion response can include attributes in the assertion. These attributes can be supplied as HTTP header variables so a client application can use them for finer grained access control.

The benefits of including attributes in HTTP headers are as follows:

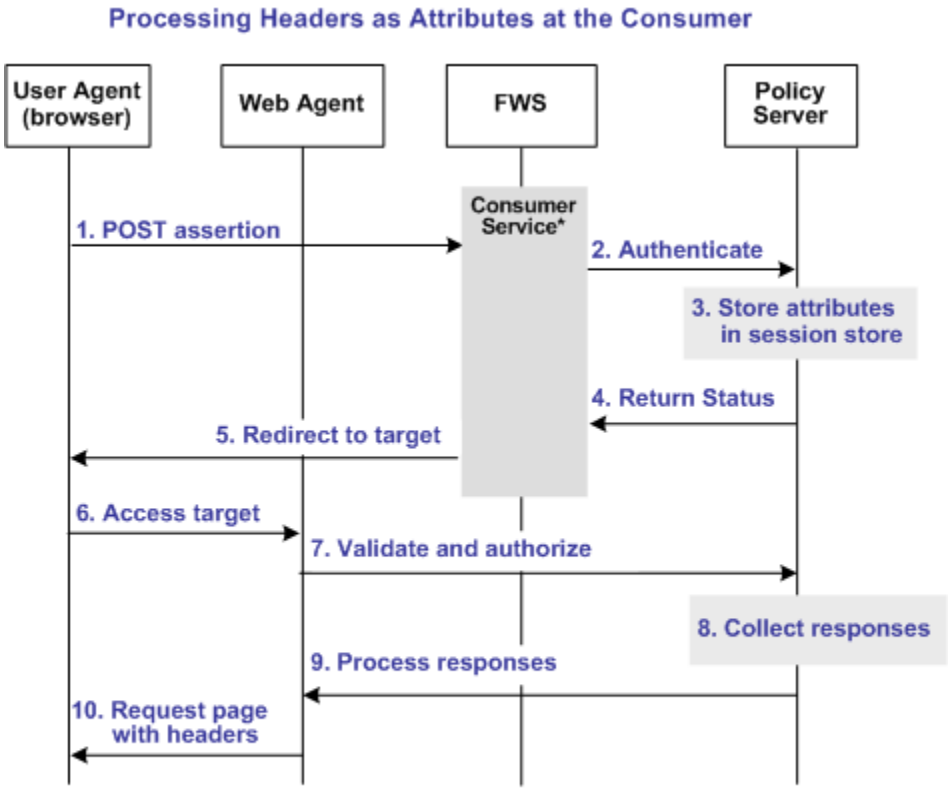
- HTTP headers are not persistent. They are present only within the request or response that contains them.
- HTTP headers, as supplied by the CA SiteMinder® Web Agent, are not visible in the browser, which reduces security concerns.

Note: The HTTP headers have size restrictions that the attributes cannot exceed. CA SiteMinder® can send an attribute in a header up to the web server size limit for a header. Only one assertion attribute per header is allowed. See the documentation for your web server to determine the header size limit.

Use Case for SAML Attributes As HTTP Headers

During authentication, a series of SAML attributes are extracted from an assertion and supplied as HTTP headers. During the authorization process, these headers are returned to the customer application.

The following flow diagram shows the sequence of events at runtime:



*Consumer service can be one of the following:
–SAML Credential Collector (SAML 1.x)
–Assertion Consumer Service (SAML 2.0)
–Security Token Consumer Service (WS-Federation)

Note: The SPS federation gateway can replace the Web Agent and Web Agent Option Pack to provide the Federation Web Services application functions. In the flow diagram, the Web Agent block would be the embedded Web Agent in the SPS federation gateway. For information about installing and configuring the SPS federation gateway, see the *Secure Proxy Server Administration Guide*.

To process the attributes as HTTP headers, the sequence of events is as follows:

1. After the assertion is generated at the asserting party, it sends the assertion to the appropriate consumer service at the relying party. The delivery mechanism (POST or Artifact or WS-Fed) is irrelevant.

   Note: The consumer service can be the SAML credential collector (SAML 1.x), the Assertion Consumer Service (SAML 2.0), or Security Token Consumer Service (WS-Federation).

2. The consumer service calls its local Policy Server to use the configured authentication scheme to authenticate the user with the assertion.

3. If the authentication scheme redirect mode parameter is set to PersistAttributes, the Policy Server caches the attributes in the session store as session variables.
4. The result of the authentication is returned to the consumer service.
5. The consumer service redirects the browser to the protected target resource.
6. The browser tries to access the target resource.
7. The Web Agent calls the Policy Server to validate the user session and to verify that the user is authorized to access the target resource.
8. The Policy Server retrieves the attributes by a configured response.
9. The Policy Server processes the responses and sends the attributes to the Web Agent.
10. The Web Agent sets the HTTP headers as necessary.

Configuration Overview to Supply Attributes as HTTP Headers

Several configuration steps are required to retrieve the SAML attributes cached in the session store and provide them as HTTP headers.

Follow these steps:

1. Select PersistAttributes as the redirect mode for the SAML authentication scheme, which enables the SAML Attributes to be returned as HTTP headers.
2. Configure an authorization rule for the realm that contains the target resource.
3. Set PersistentRealm in the realm protecting the target resource.
4. Configure a response that uses the active response type for each SAML attribute to be supplied as a header.
5. Create a policy that binds the authorization rule and active response to implement the user of attributes as HTTP headers.

Set the Redirect Mode to Store SAML Attributes

After the relying party authenticates the user with the SAML assertion, the SAML attributes are written to the session store. The browser is then redirected to the target resource.

To redirect the browser with the attribute data

1. Log in to the Administrative UI.
2. Navigate to the configuration page of the SAML authentication scheme.

3. Set the Redirect Mode parameter to Persist Attributes. Locate the Redirect Mode field as follows:

   SAML 1.x

   The Redirect Mode is in the Scheme Setup section of the main configuration page.

   SAML 2.0

   Click SAML 2.0 Configuration, SSO. The Redirect Mode is in the SSO section of the page.

   WS-Federation

   Click WS-Federation Configuration, SAML Profiles. The Redirect Mode is in the SSO section of the page.

4. Click Submit to save your changes.

The redirect mode is now set to pass on the attribute data.

Create an Authorization Rule to Validate Users

For the realm containing the protected target resource, create a rule to retrieve the SAML attributes from the session store.

The rule is based on an authorization event (onAccessAccept). The user is already authenticated by the FWS application. The Web Agent cannot reauthenticate the user and then pass on the HTTP headers. The retrieval of the attributes occurs during the authorization stage.

To create an OnAccessAccept Rule for the realm

1. Log on to the Administrative UI.
2. Navigate to Policies, Domain, Realms.
3. Select the realm with the target resource.
4. Click Create in the Rules section.

   The Create Rule page appears.

5. Enter a name and optionally, a description.
6. Enter an asterisk (*) in the Resource field.
7. Select Authorization events and OnAccessAccept in the Action section.
8. Select Enabled in the Allow/Deny and Enable/Disable section.
9. Click OK to save the rule.

The authorization rule is now defined for the realm with the protected resource.

Configure a Response to Send Attributes as HTTP Headers

Configure a response that sends the SAML attributes as HTTP headers to the Web Agent. The Web Agent processes the response and makes the header variables available to the client application.

Follow these steps:

1. Log on to the Administrative UI.
2. Navigate to Policies, Domain, Domains.
3. Select the domain for the target resource and click Modify.
4. Select the Responses tab.
5. Click Create.

   The Response dialog opens.
6. Enter a name.
7. Confirm that the Agent type is a CA SiteMinder® Web Agent.
8. Click Create Response Attribute.

   The Response Attribute dialog opens.
9. Select WebAgent-HTTP-Header-Variable in the Attribute field.
10. Select Active Response for Attribute Kind.
11. Complete the fields as follows:

    Variable Name

    Specify the name that you want for the header variable. You assign this name.

    Library Name

    smfedattrresponse

    This value must be the entry for this field.

    Function Name

    getAttributeValue

    This value must be the entry for this field.

    Parameters

    Specify the name of the attribute as it appears in the assertion.

    An agreement between you and your federated partner determines the attributes that are in the assertion.
12. Click OK to save the attribute.

13. Repeat the procedure for each attribute that is to become an HTTP header variable. You can configure many attributes for a single response.

    You return to the Response tab. The attributes that you create are listed in the Attributes List section.

14. Click OK to save the response.

    You return to the Response tab.

15. Click Submit to save the domain.

The response sends the attributes on to the Web Agent to become HTTP headers.

Create a Policy to Implement Attributes as HTTP Headers

To implement the use of SAML attributes as HTTP headers, group together the authorization event rule and active response in a policy.

Follow these steps:

1. Log on to the Administrative UI.
2. Navigate to Policies, Domain, Domains.
3. Select the domain that contains the target resource and click Modify.
4. Select the Policy tab and click Create in the Policy section.

   The Create Policy dialog opens.

5. Enter a descriptive name in the Name field.
6. Select the users who are to have access to the protected resource in the Users tab.
7. Add the authorization rule that you created previously on the Rules tab.
8. Select the authorization rule and click Add Response.

   The Available Responses dialog opens.

9. Select the active response that you created previously and click OK.

   You return to the Rules tab. The response appears with the authentication rule.

10. Click Submit to save the policy.

The policy that enables SAML attributes to be used as HTTP headers is complete.

Specify Redirect URLs for Failed SAML 2.0 Authentication

If a Service Provider cannot authenticate a user during a single sign-on transaction, that user can be redirected to a customized URL for further processing.

You can configure several optional redirect URLs for failed authentication. If the assertion is not valid, the redirect URLs allow finer control over redirecting the user. For example, if a user cannot be found in a user directory, specify a User Not Found redirect URL. This URL can send the user to a registration page.

You can configure the following URLs:

- Status redirect URLs
- HTTP error redirect URLs

Note: Configuring redirect URLs is not required.

Some of the redirect URLs are for specific status conditions. These conditions include a user is not found, the single sign-on message is invalid, or the user credentials are not accepted. Other redirect URLs handle HTTP 500, 400, 405, and 403 error conditions. If any of the conditions occur, redirect URLs can send the user to an application or a customized error page for further action.

Redirection to these customized URLs can take place only when enough information about the Identity Provider is provided to the Service Provider. For example, if during a request there is an issue in retrieving certificate information, the user is redirected to Server Error URL specified. However, if a request contains an invalid IdP ID, no redirection happens and the HTTP error code 400 is returned to the browser.

To configure optional redirect URLs

1. Navigate to the SAML 2.0 authentication scheme you want to modify.
2. Select SAML 2.0 Configuration, Advanced.
3. In the Status Redirect URLs and Modes section, fill in a URL for one or more of the fields.

   Click Help for the field descriptions.

   Federation Web Services handles the errors by mapping the authentication reason into one of the configured redirect URLs. The user can be redirected to that redirect URL to report the error.
4. Select one of the following modes:
   - 302 No Data
   - HTTP POST

5. Click OK to save your changes.

Note: These redirect URLs can be used with the Message Consumer Plug-in for further assertion processing. If authentication fails, the plug-in can send the user to one of the redirect URLs you specify.

Request Processing with a Proxy Server at the SP

When CA SiteMinder® receives certain requests at the SP, it validates the message attributes. CA SiteMinder® verifies the attributes using the local URL for Federation Web Services application. After verification, CA SiteMinder® processes the request.

For example, a logout request message can contain the following attribute:

```
Destination="http://sp.domain.com:8080/affwebservices/public/saml2slo"
```

In this example, the destination attribute in the logout message and the address of the Federation Web Services application are the same. CA SiteMinder® verifies that the destination attribute matches the local URL of the FWS application.

If the CA SiteMinder® sits behind a proxy server, the local and destination attribute URLs are not the same. The destination attribute is the URL of the proxy server. For example, the logout message can include the following destination attribute:

```
Destination="http://proxy.domain.com:9090/affwebservices/public/saml2slo"
```

The local URL for Federation Web Services, http://sp.domain.com:8080/affwebservices/public/saml2slo, does not match the Destination attribute so the request is denied.

You can specify a proxy configuration to alter how CA SiteMinder® determines the local URL used for verifying the message attribute of a request. In a proxy configuration, CA SiteMinder® replaces the <*protocol*>://<*authority*> portion of the local URL with the proxy server URL. This replacement results in a match between the two URLs.

Configure Request Processing with a Proxy Server at the SP

Specify a proxy configuration to alter how CA SiteMinder® determines the local URL used for verifying the message attribute of a request.

To use a proxy server at the Service Provider

1. Navigate to the SAML 2.0 authentication scheme you want to modify.
2. Select SAML 2.0 Configuration, Advanced.

3. In the Proxy section, enter a partial URL in the Server field. The format is *<protocol>://<authority>.*

   For example, the proxy server configuration would be:
   `http://proxy.domain.com:9090`

   If your network includes the SPS federation gateway, the Server field must specify the SPS federation gateway host and port, for example,

   `http://sps_federation_gateway.domain.com:9090`

4. Click OK to save your changes.

The Server configuration affects the URLs for the following services at the SP:

- Assertion consumer Service
- Single Logout Service

The server value becomes part of the URL CA SiteMinder® uses to verify SAML attributes, like the destination attribute.

Note: If you are using a proxy server for one URL, use it for all these URLs.

Enable Client Certificate Authentication for the Back Channel (optional)

This procedure is only for single sign-on with the artifact binding.

The Assertion Consumer Service collects information from an authentication scheme to retrieve an assertion from the Identity Provider. The scheme tells the Assertion Consumer Service what type of credentials to provide to the Identity Provider to retrieve the assertion. After the assertion is retrieved, the Identity Provider sends the assertion across a secure back channel to the Service Provider. You can use client certificate authentication to secure the back-channel.

Certificate authentication for the back-channel is optional; you can use Basic authentication instead.

To use client certificate authentication for the back channel:

1. Add a client certificate to the certificate data store (see page 264).
2. Select client certificate authentication for the back channel (see page 265). This scheme indicates that a certificate acts as credentials for the Service Provider.

   You can use non-FIPS 140 encrypted certificates to secure the back channel even if the Policy Server is operating in FIPS-only mode. However, for a strictly FIPS-only installation, use only certificates encrypted with FIPS 140-compatible algorithms.

The administrator at the asserting-side Policy Server must have configured a policy to protect the Assertion Retrieval Service. The realm for this policy must use an X.509 client certificate authentication scheme.

More information:

Configure the Authentication Scheme that Protects the Artifact Service (see page 133)

Add a Client Certificate to the Certificate Data Store

You must have a private key/certificate pair from a Certificate Authority. Add a private key/certificate pair to the certificate data store using the Administrative UI. Skip this step if the key/certificate pair is already in the data store. For instructions, see the *Policy Server Configuration Guide.*

When you import the key/certificate pair, the alias you assign must be the same value as the Name field in the authentication scheme settings. Additionally, the CN attribute of the Subject in the certificate must also match the Name field. For example, the Name is CompanyA. Therefore, the alias must be Company A, and the CN value for the Subject must read CN=CompanyA, OU=Development, O=CA, L=Islandia, ST=NY, C=US.

Important! The Name field in the authentication scheme must match the name that is assigned to the Service Provider object at the Identity Provider. If CA SiteMinder® is the Identity Provider, the Name in the authentication scheme must match the Name field in the General settings of the object.

Configure the Client Certificate Option for the Back Channel

If you enable client certificate authentication for the back channel, the certificate serves as your credential.

To present a client certificate as a credential

1. Navigate to the SAML 2.0 authentication scheme.
2. Select SAML 2.0 Configuration, SSO.

   The SSO page displays.
3. Select HTTP-Artifact in the Bindings section.
4. Click OK.
5. Move to the Encryption & Signing page.
6. In the Backchannel section, select Client Cert for the Authentication field.
7. Fill in a value for the SP Name.
8. Click OK.

How To Protect Resources with a SAML 2.0 Authentication Scheme

Protect target federation resources by configuring a CA SiteMinder® policy that uses the SAML 2.0 authentication scheme.

To protect a federation resource with a SAML authentication scheme:

1. Create a realm that uses the SAML authentication scheme. The realm is the collection of target resources that users request.

   Create a realm in one of the following ways:

   - Create a unique realm (see page 170) for each authentication scheme already configured.
   - Configure a single target realm (see page 170) that uses a custom authentication scheme to dispatch requests to the corresponding SAML authentication schemes. Configuring one realm with a single target for all Identity Providers simplifies configuration of realms for SAML authentication.

2. After you configure a realm, establish an associated rule and optionally, a response.
3. Group the realm, rule, and response into a policy that protects the target resource.

Important! Each target URL in the realm is also identified in an unsolicited response URL. An unsolicited response is sent from the Identity Provider to the Service Provider, without an initial request from the Service Provider. The unsolicited response contains the target. At the Identity Provider, an administrator must include this response in a link so the Identity Provider can redirect the user to the Service Provider.

Configure a Unique Realm for Each Authentication Scheme

The procedure for configuring a unique realm for each SAML or WS-Federation authentication scheme follows the standard instructions for creating realms.

Follow these steps:

1. Navigate to Policy, Domain, Domains.

   The page to create domains displays.
2. Click Create Domain.
3. Enter a domain name.
4. Add the user directory to the domain. This directory is the one that contains the users requesting access to federated resources.
5. Select the Realm tab and create a realm.
   - In the Agent field, select the Web Agent protecting the web server where the target resources reside.
   - Select the appropriate authentication scheme in the Authentication Scheme field.
6. Create a rule for the realm.

   As part of the rule, select an action (Get, Post, or Put) that allows you to control processing when users authenticate.
7. Select the Policies tab and configure a policy that protects the target federation resource. Associate the realm that you previously created with this policy.

A policy with a unique realm now protects the federated resources.

Configure a Single Target Realm for All Authentication Schemes

To simplify configuration of realms for authentication schemes, create a single target realm for multiple sites generating assertions.

To do this task, set up the following components:

- A single custom authentication scheme

  This custom scheme forwards requests to the corresponding SAML or WS-Federation authentication schemes that you already configured for each asserting party.

- A single realm with one target URL

Create Authentication Schemes for the Single Target Realm

To define a custom authentication scheme for a single target realm, you must:

- Configure the authentication schemes.
- Define a parameter in the custom scheme that tells the Policy Server which authentication schemes to apply to resource requests.

First, verify that there are configured SAML or WS-Federation authentication schemes. If not, configure these schemes that the custom scheme can reference.

To create the authentication scheme

1. Navigate to Infrastructure, Authentication, Authentication Schemes.

   The Create Authentication Scheme page appears.

2. Create one or more authentication schemes according to the procedures for the protocol you are using.
3. Click OK to exit.

More information:

SAML 1.x Authentication Schemes (see page 148)
WS-Federation Authentication Scheme Overview (see page 297)
How to Configure a SAML 2.0 Authentication Scheme (see page 235)

Create the Custom Authentication Scheme

A single target realm relies on a specific custom authentication scheme to work properly.

To configure a custom authentication scheme for a single target realm

1. Navigate to Infrastructure, Authentication, Authentication Schemes.

   The Create Authentication Scheme page appears.

2. Complete the fields as follows:

   Name

   Enter a descriptive name for the custom authentication scheme, such as SAML Custom Auth Scheme.

3. In the Scheme Common Setup section, complete the following fields:

   Authentication Scheme Type

   Custom Template

   Protection Level

   Accept the default of set a new level.

4. In the Scheme Setup section, complete the following fields:

   Library

   smauthsinglefed

   Secret

   Leave this field blank.

   Confirm Secret

   Leave this field blank.

   Parameter

   Specify one of the following parameters:

   - SCHEMESET=LIST; <*saml-scheme1*>;<*saml_scheme2*>

     Specifies the list of SAML authentication scheme names to use. If you configured an artifact scheme named artifact_producer1 and POST profile scheme named samlpost_producer2, you enter these schemes. For example:

     SCHEMESET=LIST;artifact_producer1;samlpost_producer2

   - SCHEMESET=SAML_ALL;

     Specifies all the configured schemes. The custom authentication scheme enumerates all the SAML authentication schemes and finds the one with the correct Provider Source ID for the request.

- SCHEMESET=SAML_POST;

  Specifies all the SAML POST Profile schemes that you have configured. The custom authentication scheme enumerates the POST Profile schemes and finds the one with the correct Provider Source ID for the request.

- SCHEMESET=SAML_ART;

  Specifies all the SAML artifact schemes that you have configured. The custom authentication scheme enumerates the artifact schemes and finds the one with the correct Provider Source ID for the request.

- SCHEMESET=WSFED_PASSIVE;

  Specifies all the WS-Federation authentication schemes to find the one with the correct Account Partner ID.

Enable this scheme for CA SiteMinder® Administrators

Leave unchecked.

5. Click Submit.

The custom authentication scheme is complete.

Configure the Single Target Realm

After you configure the authentication schemes and associate them with a custom scheme, configure a single target realm for federation resources.

Follow these steps:

1. Navigate to Policies, Domain, Domains.
2. Modify the policy domain for the single target realm.
3. Select the Realms tab and click Create.

   The Create Realm dialog opens.

4. Enter the following values to create the single target realm:

   Name

   Enter a name for this single target realm.

5. Complete the following field in the Resource option:

   Agent

   Select the Web Agent protecting the web server with the target resources.

   Resource Filter

   Specify the location of the target resources. The location is where any user requesting a federated resource gets redirected.

   For example, /FederatedResources.

6. Select the Protected option in the Default Resource Protection section.
7. Select the previously configured custom authentication scheme in the Authentication Scheme field.

   For example, if the custom scheme was named Fed Custom Scheme, you would select this scheme.
8. Click OK.

The single target realm task is complete.

Configure the Rule for the Single Target Realm

After you configure the single target realm, configure a rule to protect the resources.

1. Navigate to the Modify page for the single target Realm.
2. Click Create in the Rules section.

   The Create Rule page appears.
3. Enter values for the fields on the rules page.
4. Click OK.

The single target realm configuration includes the new rule.

Create a Policy Using the Single Target Realm

Create a policy that references the single target realm. Remember that the single target realm uses the custom authentication scheme that directs requests to the appropriate SAML authentication scheme.

Note: This procedure assumes that you have already configured the domain, custom authentication scheme, single target realm and associated rule.

Follow these steps:

1. Navigate to the previously configured domain.
2. Select the Policies tab and click create.

   The Create Policy page opens.
3. Enter a name and a description of the policy in the General section.
4. Add users to the policy from the Users section.
5. Add the rule that you created for the single target realm from the Rules tab.

   The remaining tabs are optional.

6. Click OK.
7. Click Submit.

The policy task is complete. When a request triggers this policy, it relies on the single realm and associated authentication schemes to authenticate the user.

Chapter 16: Configure a WS-Federation Account Partner

Prerequisites for an Asserting Partner(legacy)

To configure an asserting partner, verify the following conditions:

- The Policy Server is installed.
- One of the following options is installed:
  - The Web Agent and the Web Agent Option Pack. The Web Agent authenticates users and establishes a CA SiteMinder® session. The Option Pack provides the Federation Web Services application. Be sure to deploy the FWS application on the appropriate system in your network.
  - The SPS federation gateway has an embedded Web Agent and the Federation Web Services application on the embedded Tomcat web server.

  For more information, see the *Web Agent Option Pack Guid*e.
- Private keys and certificates are imported for functions that require signing and decrypting messages.
- A relying partner is set up within the federated network.

How To Configure an Account Partner

CA SiteMinder®, as an Account Partner generates assertions for its business partners, the Resource Partners. To establish a federated partnership, the Account Partner needs information about each Resource Partner. Create a Resource Partner object for each partner. Define how the two entities communicate to pass assertions and to satisfy profiles, such as single sign-on.

Follow these steps:

1. Create a Resource Partner object.
2. Add the Resource Partner to the affiliate domain.
3. Specify the general identifying information for the Resource Partner.
4. Select users from a user store. The Account Partner generates assertions for the users you select.

5. Specify the Name ID to include in the assertion.
6. Configure the single sign-on profile.

   You can save a Resource Partner entity without configuring a complete SSO profile. However, you cannot pass an assertion to the Resource Partner without configuring SSO.
7. Complete optional configuration tasks (see page 274).

Tips:

- Certain parameter values at the Account Partner and the Resource Partner must match for the configuration to work. A list of those parameters can be found in Configuration Settings that Must Use the Same Values (see page 361).
- Use the correct URLs for the Federation Web Services servlets. A list of URLs can be found in Federation Web Services URLs Used in SiteMinder Configuration (see page 367)

Optional Configuration Tasks for an Account Partner

The optional tasks for configuring a Account Partner include:

- Configure single sign-on restrictions:
  - Configure time restrictions (see page 280) for Resource Partners.
  - Set IP address restrictions (see page 280) to limit the addresses that are used to access Resource Partners.
- Configure attributes for inclusion in assertions (see page 291).
- Configure sign out (see page 290).
- Customize a SAML response (see page 142) using the Assertion Generator plug-in.

Navigating Legacy Federation Dialogs

The Administrative UI provides two ways to navigate to the legacy federation configuration dialogs.

You can navigate in one of two ways:

- Following a wizard to configure a new legacy federation object.

  When you create an object, a page displays with a configuration wizard. Follow the steps in the configuration wizard to create the object.

- Selecting tabs to modify an existing legacy federation object.

  When you modify an existing object, a page displays with a series of tabs. Modify the configuration from these tabs. These tabs are the same as the steps in the configuration wizard.

Add a Resource Partner to an Affiliate Domain

To identify a Resource Partner as an available consumer of assertions, add the Resource Partner to an affiliate domain at the Account Partner. You then configure the Resource Partner so that the Account Partner can issue security token response messages containing assertions.

Follow these steps:

1. Navigate to Federation, Legacy Federation, Resource Partners.
2. Click Create Resource Partner.

   The Create Resource Partner page appears.
3. Select an affiliate domain, then click Next.

   The General page appears.
4. Fill in the fields at the top of the dialog.

   Click Help for field descriptions.
5. Select Enabled so the Account Partner can recognize the configured Resource Partner.

Configure General Information for the Resource Partner Object

Select the General page to name the Resource Partner and provide details, such as the Resource Partner and Account Partner IDs. In addition, you can configure IP address and time restrictions for accessing a Service Provider.

To configure the general settings

1. Navigate to the General settings.

2. Fill in values for the fields, noting the required fields.

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

   Note the following information about the Skew Time field.

   Skew Time

   Specifies the number of seconds subtracted from the current system time. This calculation accounts for Resource Partners with clocks that are not synchronized with the Account Partner.

   For single sign-on, the value of the skew time and the single sign-on validity duration determine how long an assertion is valid. Review how the assertion validity (see page 128) is calculated to understand more about the skew time.

3. For debugging purposes only, you can temporarily disable all signature processing (both signing and verification of signatures) by selecting the Disable Signature Processing checkbox.

   Important! Signature processing is enabled by default because the WS-Federation Passive Requester profile for single sign-on requires it.

Authenticate Users with No CA SiteMinder® Session

When you add a Resource Partner to an affiliate domain, one of the parameters you are required to set is the Authentication URL parameter.

The Authentication URL points to the redirect.jsp file. This file is installed at the Account Partner site where you install the Web Agent Option Pack or the SPS federation gateway. Protect the redirect.jsp file with a CA SiteMinder® policy. The policy triggers an authentication challenge to users who request a protected Resource Partner resource but do not have a CA SiteMinder® session.

A CA SiteMinder® session is required for the following bindings:

- For single sign-on using an HTTP POST binding

  A user must have a session; however, the session does not have to be persistent. Assertions are delivered directly to the Resource Partner through the browser. The assertions do not have to be stored in the session store.

- For signout

  If you enable single logout, a persistent session is required. When a user first requests a resource, the Account Partner stores the session in the session store. The session information is necessary when a single logout is later executed.

After a user is authenticated and successfully accesses the redirect.jsp file, a session is established. The redirect.jsp file redirects the user back to the Account Partner Web Agent or the SPS federation gateway. CA SiteMinder® then processes the request.

The procedure for protecting the Authentication URL is the same regardless of the following deployments:

- Web Agent Option Pack installed on the same system as the Web Agent
- Application server with a Web Agent installed on a web server proxy
- Application server with an Application Server Agent
- SPS federation gateway that is installed at the Identity Provider

Configure a Policy to Protect the Authentication URL

To protect the Authentication URL

1. Log in to the Administrative UI.
2. Create Web Agents to bind to the realms that you define for the asserting party web server. Assign unique agent names for the web server and the FWS application or use the same agent name for both.
3. Create a policy domain for the users who are challenged when they try to access a consumer resource.
4. Select the users that must have access to the resources that are part of the policy domain.
5. Define a realm for the policy domain with the following values:

   Agent

   Agent for the asserting party web server

   Resource Filter

   Web Agents r6.x QMR 6, r12.0 SP2, r12.0 SP3 and SPS federation gateway enter:

   /siteminderagent/redirectjsp/

   The resource filter /siteminderagent/redirectjsp/ is an alias that the FWS application sets up automatically. The alias references include:

   - Web Agent:

     web_agent_home/affwebservices/redirectjsp

   - SPS federation gateway:

     sps_home/secure-proxy/Tomcat/webapps/affwebservices/redirectjsp

   Persistent Session

   For the SAML artifact profile only, select the Persistent check box in the Session section of the realm dialog. If you do not configure a persistent session, the user cannot access consumer resources.

   For the remaining settings, accept the defaults or modify as needed.

6. Click OK to save the realm.

7. Create a rule for the realm. In the Resource field, accept the default value, the asterisk (*), to protect all resources for the realm.
8. Create a policy for the asserting party web server that includes the rule created in the previous step.
9. Complete the task Select Users for Which Assertions are Generated (see page 123).

Assertion Validity for Single Sign-on

For single sign-on, the values of the Skew Time and the Validity Duration determine how CA SiteMinder® calculates the total time that an assertion is valid. CA SiteMinder® applies the skew time to the generation and consumption of assertions.

Note: In this description, the asserting party is the SAML 1.x Producer, SAML 2.0 Identity Provider, or WS-Federation Account Partner. The relying party is the SAML 1.x Consumer, the SAML 2.0 Service Provider, or the WS-Federation Resource Partner.

In the assertion document, the NotBefore and NotOnOrAfter values represent the beginning and end of the validity interval.

At the asserting party, CA SiteMinder® sets the assertion validity. The validity interval is the system time when the assertion is generated. CA SiteMinder® sets the IssueInstant value in the assertion using this time then subtracts the skew time value from the IssueInstant value. The resulting time is the NotBefore value.

NotBefore=IssueInstant - Skew Time

To determine the end of the validity interval, CA SiteMinder® 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, CA SiteMinder® performs the same calculations as it does at the asserting party to determine if the assertion it receives is valid.

Calculating Assertion Validity with CA SiteMinder® at Both Sides of the Partnership

If CA SiteMinder® is at both sides of a partnership, the assertion validity is the sum of the validity duration plus two times the skew time. The equation is:

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

The initial part of the equation (2 x Skew Time + Validity Duration) represents the beginning and end of 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 legacy federation, the Validity Duration is only set at the asserting party.

Example

Asserting Party

The values at the asserting party are as follows:

- IssueInstant=5:00PM
- Validity Duration=60 seconds
- Skew Time = 60 seconds
- NotBefore = 4:59PM
- NotOnOrAfter=5:02PM

Relying Party

The relying party uses the NotBefore and NotOnOrAfter values from the assertion and applies its skew time to those values. This formula is how the relying party calculates new NotBefore and NotOnOrAfter values.

- Skew Time = 180 seconds (3 minutes)
- NotBefore = 4:56PM
- NotOnOrAfter=5:05PM

Assertion Validity Window

Using the values in this example, the calculation for the total assertion validity window is:

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

Configure Time Restrictions for Resource Partner Availability (optional)

You can specify time restrictions for Resource Partner resource availability. When you specify a time restriction, access to the Resource Partner resources is only during the period specified. If a user tries accessing a resource outside of the designated period, the Account Partner does not generate a SAML assertion.

Note: Time restrictions are based on the system clock of the server on which the Policy Server is installed.

To specify a time restriction

1. Begin at the General settings.

   In the Restrictions section of the page, click Set in the Time section.

   The Time Restriction page displays.

2. Complete the schedule. This schedule grid is identical to the Time Restriction grid for rule objects. For more information, see the *Policy Server Configuration Guide*.

3. Click OK.

The time restriction schedule is set.

Configure IP Address Restrictions for Resource Partners (optional)

You can specify an IP address, range addresses, or a subnet mask of the web server to access a Resource Partner. If IP addresses are specified for a Resource Partner, the Resource Partner only accepts users from the appropriate IP addresses.

To specify IP addresses

1. Begin at the General settings.

   In the Restrictions section of the page, click Add in the IP Address area.

   The IP Restrictions page appears.

2. Select the option for the type of IP address you are adding, then complete the associated fields for that address type.

   Note: If the IP address is unknown but you have a domain name for the address, click the DNS Lookup button. This button opens the DNS Lookup page. Enter a fully qualified host name in the Host Name field and click OK.

   - Single Host--specifies a single IP address that hosts the browser. If you specify a single IP address, users can access the Resource Partner only from the specified IP address.
   - Host Name--specifies a web server using its host name. If you specify a host name, the Resource Partner is only accessible to users from the specified host.

- Subnet Mask--specifies a subnet mask for a web server. If you specify a subnet mask, the Resource Partner is only accessible to users from the specified subnet mask. If you select this button, the Add An Address and Subnet Mask dialog opens. Use the Left and Right arrow buttons, or click and drag the slider bar to select a subnet mask.
- Range--specifies IP address range. If you specify a range of IP addresses, the Resource Partner only permits users from one of the IP addresses in the range of addresses. Enter a starting (FROM) and ending (TO) addresses to determine the range.

3. Click OK to save your configuration.

Select Users for Which Assertions are Generated

As part of the configuration at the asserting party, include a list of users and groups for which the Assertion Generator generates SAML assertions. The asserting party is either a SAML 1.x Producer, a SAML 2.0 Identity Provider, or a WS Federation Account Partner.

You can only add users and groups from directories that are in an affiliate domain.

To specify users and groups for federated transactions

1. Navigate to the Users settings for the partner you are configuring.

   The User Directories page displays entries for each user directory for the policy domain.

2. Add users or groups from the user directory to the policy.

   In each user directory table, you can select Add Members, Add Entry, Add All. Depending on which method you select, a dialog opens enabling you to add users.

   - 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.
   - If you select Add Entry, select users by manual entry (see page 125) in the User Directory Search Expression Edit dialog.

   Edit or delete a user or group by clicking the right arrow (>) or minus sign (-), respectively.

3. Select individual users, user groups, or both using whatever method and click OK.

   The User Directories page reopens and lists the new users in the user directory table.

More information:

Exclude a User or Group from Access to a Resource (see page 124)
Allow Nested Groups Access to Resources (see page 125)
Add Users by Manual Entry (see page 125)

Exclude a User or Group from Access to a Resource

You can exclude users or groups of users from obtaining an assertion.

Follow these steps:

1. Navigate to the User settings.
2. Select a user or group from the list for a particular user directory.
3. Click Exclude to exclude the selected user or group.

   The selection is reflected in the Administrative UI.
4. Click OK to save your changes.

Allow Nested Groups Access to Resources

LDAP user directories can contain groups that have subgroups. In complex directories, groups nesting in a hierarchy of other groups is one way to organize large amounts of user information.

If you enable a search for users in nested groups, any nested group is searched for the requested user record. If you do not enable nested groups, the Policy Server only searches the group you specify.

To enable searching in nested groups

1. Navigate to the Users settings.

   If the associated affiliate domain contains more than one user directory, each user directory appears in its own section.
2. Select the Allow Nested Groups check box to enable searching within nested groups.

Add Users by Manual Entry

When you specify users for assertion generation, one of the options is to identify users by manual entry.

Follow these steps:

1. Navigate to the Users settings for the partner you are configuring.

   If the affiliate domain contains more than one user directory, all the directories appear on the User Directories page.

2. Click Add Entry.

   The User Directory Search Express Edit page displays.

3. Select the search option then complete the fields for that search option.

   Where to Search

   For LDAP directories, select an option from the drop-down list:

   Validate DN

   LDAP search locates this DN in the directory.

   Search Users

   LDAP search is limited to matches in user entries.

   Search Groups

   LDAP search is limited to matches in group entries.

   Search Organizations

   LDAP search is limited to matches in organization entries.

   Search Any Entry

   LDAP search is limited to matches in user, group, and organization entries.

- For Microsoft SQL Server, Oracle and WinNT directories, you can enter a user name in the Manual Entry field.
- For a Microsoft SQL Server or Oracle, you can enter a SQL query instead. For example:

  SELECT NAME FROM EMPLOYEE WHERE JOB =’MGR’;

  The Policy Server performs the query as the database user specified in the Username field of the Credentials and Connection tab for the user directory. When constructing the SQL statement for the Manual Entry field, be familiar with the database schema for the user directory. For example, if you are using the SmSampleUsers schema and you want to add specific users, select a user entry from the SmUser table.
- For an LDAP directory, enter **all** in the Manual Entry field to add all directory entries.

4. Click OK to save your changes.

Configure a Name ID for a WS-Federation Assertion

A name ID specifies a user in an assertion in a unique way. The value you configure in the Administrative UI is included in the assertion sent to the Resource Partner.

The format of the name ID establishes the type of content that is used for the ID. For example, if the format is the User DN, the content is a UID.

To configure a name ID

1. Navigate to the Resource Partner object you want to configure.
2. Select the Name IDs settings.
3. Select the Name ID Format.

   For a description of each format, see the *Assertions and Protocols for the OASIS Security Assertion Markup Language (SAML) V2.0* specification.
4. Select the Name ID Type from the following options:
   - Static value
   - User attribute
   - DN attribute (with or without nested groups)

   The contents of the Name ID Fields change according to the Name ID Type.
5. Complete the Name ID fields for the selected Name ID Type.

Configure Single Sign-on for WS-Federation

An assertion provides the necessary identity information to facilitate single sign-on at the Resource Partner. The Account Partner generates a SAML 1.1 assertion for a user with an established session. The Account Partner places the assertion in a WS-Federation RequestSecurityTokenResponse message then delivers the token to the Resource Partner. The Resource Partner consumes security tokens and establishes a session that is based on the contents of the WS-Federation security token.

As part of single sign-on configuration, determine how the Account Partner delivers an assertion to a Resource Partner.

To configure single sign-on at the Account Partner

1. Navigate to the SAML Profiles settings for the Resource Partner object.
2. Complete the fields in the SSO section of the page.

   Click Help for field descriptions.
3. Click Submit to save your changes.

Initiate Single Sign-on at the Account Partner

A user can visit the Account Partner before going to the Resource Partner. If the user goes to the Account Partner first, a link must generate an HTTP Get request. The hard-coded link points to the Single Sign-on Service of the Account Partner. The request contains the RP Provider ID and optionally other parameters.

The syntax for the link to the Single Sign-on Service is as follows:

https://*ap_server:port*/affwebservices/public/wsfedsso?wa=wsignin1.0&wtrealm=*RP_ID*

ap_server:port

Specifies the server and port number of the system at the Account Partner. The system is hosting the Web Agent Option Pack or the SPS federation gateway, depending on which component is installed in your federation network.

RP_ID

Resource Partner identity, The entity ID is case-sensitive. Enter it exactly as it appears in the Administrative UI.

Initiate Single Sign-on at the Resource Partner

When a user starts at the Resource Partner to initiate single sign-on, typically the user selects from a list of Account Partners. The site selection page is in an unprotected realm.

The link on the site selection page points to the Single Sign-on Service at an Account Partner. After the link is selected, the Resource Partner redirects the user to the Account Partner to get the assertion.

Customize a SAML Assertion Response (optional)

You can modify the assertion content using an assertion generator plug-in. The plug-in enables you to customize the content of an assertion using the business agreements between you and your partners and vendors. One plug-in is allowed for each partner.

The steps to configure an assertion generator plug-in are:

1. Install the CA SiteMinder® SDK, if you have not done so already.
2. Implement the AssertionGeneratorPlugin.java interface, which is part of the SDK.
3. Deploy your assertion generator plug-in implementation class.
4. Enable the assertion generator plug-in parameters in the Administrative UI.

Additional information about the Assertion Generator plug-in can be found as follows:

- Reference information (method signatures, parameters, return values, data types), and also the new constructor for UserContext class, are in the *Javadoc Reference*. Refer to the AssertionGeneratorPlugin interface in the Javadoc.
- Overview and conceptual information for authentication and authorization APIs is in the *CA SiteMinder® Programming Guide for Java*.

Implement the AssertionGeneratorPlugin Interface

The first step in creating a custom assertion generator plug-in is to implement the AssertionGeneratorPlugin interface.

Follow these steps:

1. Provide a public default constructor method that contains no parameters.
2. Provide code so that the implementation is stateless. Many threads must be able to use a single plug-in class.

3. Implement methods in the interface to satisfy your requirements.

   The implementation must include a call to the customizeAssertion methods. You can overwrite the existing implementations. See the following sample classes for examples:

   SAML 1.x/WS-Federation

   AssertionSample.java

   SAML 2.0

   SAML2AssertionSample.java

   The sample classes are located in the directory /sdk/samples/assertiongeneratorplugin.

   The contents of the parameter string that your implementation passes into the customizeAssertion method is the responsibility of the custom object.

Deploy the Assertion Generator Plug-in

After you have coded your implementation class for the AssertionGeneratorPlugin interface, compile it and verify that CA SiteMinder® can find your executable file.

To deploy the assertion generator plug-in

1. Compile the assertion plug-in Java file.

   Compilation requires the following .jar files, which are installed with the Policy Server:

   - *policy_server_home*/bin/jars/SmJavaApi.jar
   - *policy_server_home*/bin/thirdparty/xercesImpl.jar
   - *policy_server_home*/bin/endorsed/xalan.jar

2. In the JVMOptions.txt file, modify the -Djava.class.path value so it includes the classpath for the plug-in. This modification enables the plug-in to be loaded with the modified classpath. Locate the JVMOptions.txt file in the directory *installation_home*\siteminder\config.

   Note: Do not modify the classpath for xercesImpl.jar, xalan.jar, or SMJavaApi.jar.

3. Enable the plug-in.

Enable the Assertion Generator Plug-in

After writing an assertion generator plug-in and compiling it, enable the plug-in by configuring settings in the Administrative UI. The UI parameters let CA SiteMinder® know where to find the plug-in.

Do not configure the plug-in settings until you deploy the plug-in (see page 143).

Follow these steps:

1. Log on to the Administrative UI.
2. Click Federation, Legacy Federation, Resource Partners.
3. Select an existing Resource Partner entry or create one.
4. Navigate to the General settings.
5. In the Advanced section, complete the following fields:

   Java Class Name

   Specify a Java class name for an existing plug-in

   The plug-in class can parse and modify the assertion, and then return the result to the Assertion Generator for final processing.

   Only one plug-in is allowed for each partner. For example, com.mycompany.assertiongenerator.AssertionSample

   Parameter

   (Optional) Specify a string of parameters that is passed to the plug-in specified in the Java Class Name field.

   Note: Instead of enabling the assertion plug-in through the Administrative UI, you can use the Policy Management API (C or Perl) to integrate the plug-in. For more information, see the *CA SiteMinder® Programming Guide for C* or the *CA SiteMinder® Programming Guide for Java*.

6. Restart the Policy Server.

   Restarting the Policy Server ensures that the latest version of the assertion plug-in is picked up after being recompiled.

Customize the Assertion with Attributes from a Web Application

You can use an assertion generator plug-in to add web application attributes tp an assertion. This is another way to customize the assertion.

To include web application attributes in an assertion

1. Compile the assertion plug-in Java file.

   Compilation requires the following .jar files, which are installed with the Policy Server:

   - *policy_server_home*/bin/jars/SmJavaApi.jar
   - *policy_server_home*/bin/thirdparty/xercesImpl.jar
   - *policy_server_home*/bin/endorsed/xalan.jar

2. In the JVMOptions.txt file, modify the -Djava.class.path value so it includes the classpath for the plug-in. This modification enables the plug-in to be loaded with the modified classpath. Locate the JVMOptions.txt file in the directory *installation_home*\siteminder\config.

   Note: Do not modify the classpath for xercesImpl.jar, xalan.jar, or SMJavaApi.jar.

3. Configure a sample plug-in.

   An APIContext class in the SMJavaAPI has a new method, getAttrMap(), which returns a map object containing the attributes from the web application included in the assertion. In the SiteMinder SDK, there are two sample Assertion Generator plug-ins that show how to use this map object:

   - SAML2AppAttrPlugin.java (SAML 2.0)
   - WSFedAppAttrPlugin.java (WS-Federation)

   These samples are located in the directory sdk/samples/assertiongeneratorplugin. They enable the Assertion Generator to add attributes from a web application to an assertion.

4. Log in to the Administrative UI.
5. Select Federation, Legacy Federation, SAML Service Providers or Resource Partners.
6. Select an exiting entry or create one.
7. Navigate to the General settings.

8. In the Assertion Generator Plug-in section, complete the following fields:

   Java Class Name

   Names the Java class for the plug-in. For example, the sample classes included with the CA SiteMinder® SDK are:

   - com.ca.assertiongenerator.SAML2AppAttrPlugin

     (SAML 2.0)

   - com.ca.assertiongenerator.WSFedAppAttrPlugin

     (WS-Federation)

   Parameter

   Specify a string of parameters that is passed to the plug-in specified in the Java Class Name field. These parameters would be the attributes that you want to include in the assertion.

   Note: Instead of configuring the settings through the Administrative UI, you can use the Policy Management API (C or Perl) to integrate the plug-in. For instructions, see the *CA SiteMinder® Programming Guide for C* or the *CA SiteMinder® Programming Guide for Java*.

9. Restart the Policy Server.

   Restarting the Policy Server verifies that the latest version of the assertion plug-in is picked up after being recompiled.

Configure Signout for WS-Federation

Signout is the process of a user being logged out of all sessions for the browser that initiated the logout. Signout does not necessarily end all sessions for a user. For example, if the user has two browsers open, that user can establish two independent sessions. Only the session for the browser that initiates the signout is terminated at all federated sites for that session. The session in the other browser is still active.

A user can initiate a signout request from an Account Partner or a Resource Partner. The request is triggered by clicking a link pointing to the appropriate servlet.

Note: The system only supports the WS-Federation Passive Request for sign out.

By configuring the settings in the Signout section, you are informing the Account Partner how the Resource Partner supports signout.

If you enable signout, you must also:

- Enable the session store at the Account Partner using the Policy Server Management Console.

  For information about the session store, see the *Policy Server Administration Guide*.

- Sign-out requires a valid SiteMinder persistent session. which is established during Single Sign-on. Configure persistent sessions for the realm with the protected resources, including the authentication URL, at the Resource Partner.

  For information about realms, see the *Policy Server Configuration Guide*.

To configure signout

1. Navigate to the SAML Profiles page for the Resource Partner you want to configure.
2. In the Signout section, select Enable Signout.
3. Enter values for the following URL fields:
   - Signout Cleanup URL
   - Signout Confirm URL

   These fields must each have an entry that starts with https:// or http://.

   Click Help for field descriptions.
4. Click OK.

Configure Attributes for WS-Federation Assertions (optional)

Attributes can provide information about a user requesting access to a Resource Partner resource. An attribute statement passes user attributes, DN attributes, or static data from the Account Partner to the Resource Partner in a SAML assertion. Any configured attributes are included in the assertion in one <AttributeStatement> element or the <EncryptedAttribute> element in the assertion.

Note: Attribute statements are not required in an assertion.

Servlets, web applications, or other custom applications use attributes to display customized content or enable other custom features. When used with web applications, attributes can implement fine-grained access control by limiting user activity at the Resource Partner. For example, you can send an attribute variable named Authorized Amount set to a maximum dollar amount. The amount is the limit that the user can spend at the Resource Partner.

Attributes take the form of name/value pairs. When the Resource Partner receives the assertion, it makes the attribute values available to applications.

Attributes can be made available as HTTP Headers or HTTP Cookies.

The HTTP headers and HTTP cookies have size restrictions that assertion attributes cannot exceed. The size restrictions are as follows:

- For HTTP headers, CA SiteMinder® can send an attribute in a header up to the web server size limit for a header. Only one assertion attribute per header is allowed. See the documentation for your web server to determine the header size limit.
- For HTTP cookies, CA SiteMinder® can send a cookie up to the size limit for a cookie. Each assertion attribute is sent as its own cookie. The cookie size limit is browser-specific, and that limit is for all attributes being passed to the application, not for each attribute. See the documentation for your web browser to determine the cookie size limit.

Configure Assertion Attributes for WS-Federation

To configure assertion attributes

1. Navigate to the Attributes page for the Resource Partner object you are configuring.
2. Click Add in the Attributes section.

   The Add Attributes dialog appears.
3. From the Attribute drop-down, select the name format identifier. The <NameFormat> attribute in the <Attribute> element of an assertion specifies the identifier. This value classifies the attribute name so that the Resource Partner can interpret the name.

   The options are:

   - EmailAddress
   - UPN
   - CommonName
   - Group
   - NameValue

   For more information about these options, see the WS-Federation specification.

4. In the Attribute Setup section, select one of the following options:
   - Static
   - User Attribute
   - DN Attribute

   The selection of the following option determines the available fields in the Attribute Fields section.

   Click Help for field descriptions.
5. Optional. The attribute can be retrieved from an LDAP user directory with nested groups. For the Policy Server to retrieve DN attributes from the nested groups, select the Allow Nested Groups check box in the Attribute Kind section.
6. Complete the necessary fields for you Attribute Kind and save the changes.

Specify the Maximum Length of Assertion Attributes

The maximum length for user assertion attributes is configurable. To modify the maximum length of assertion attributes, change the settings in the EntitlementGenerator.properties file.

The property name in the file is specific to the protocol you are configuring.

Follow these steps:

1. On the system where the Policy Server is installed, navigate to *policy_server_home*\config\properties\EntitlementGenerator.properties.
2. Open the file in a text editor.

3. Adjust the maximum user attribute length for the protocols in use in your environment. The settings for each protocol are as follows:

   WS-Federation

   Property Name:
   com.netegrity.assertiongenerator.wsfed.MaxUserAttributeLength

   Property Type: Positive Integer value

   Default Value: 1024

   Description: Indicates the maximum attribute length for WS-FED assertion attributes.

   SAML 1.x

   Property Name:
   com.netegrity.assertiongenerator.saml1.MaxUserAttributeLength

   Property Type: Positive Integer value

   Default Value: 1024

   Description: Indicates the maximum attribute length for SAML1.1 assertion attributes.

   SAML 2.0

   Property Name:
   com.netegrity.assertiongenerator.saml2.MaxUserAttributeLength

   Property Type: Positive Integer value

   Default Value: 1024

   Description: Indicates the maximum attribute length for SAML2.0 assertion attributes

4. Restart the Policy Server after any change to these parameters.

Use a Script to Create a New Attribute

The Advanced section of the Attribute dialog contains the Script field. This field displays the script that CA SiteMinder® generates based on your entries in the Attribute Setup section. You can copy the contents of this field and paste them into the Script field for another response attribute.

Note: If you copy and paste the contents of the Script field for another attribute, select the appropriate option in the Attribute Kind section.

Chapter 17: Configure CA SiteMinder® as a WS-Federation Resource Partner

This section contains the following topics:

Prerequisites for a Relying Partner

For CA SiteMinder® to act as the relying partner, complete following tasks:

- Install the Policy Server.
- Install one of the following components:
  - The Web Agent and the Web Agent Option Pack. The Web Agent authenticates users and establishes a session. The Option Pack provides the Federation Web Services application. Be sure to deploy the FWS application on the appropriate system in your network.
  - The SPS federation gateway, which has an embedded Web Agent and has the Federation Web Services application on the embedded Tomcat web server.

  For more information, see the *Web Agent Option Pack Guid*e.
- Private keys and certificates are imported for functions that require verification and encrypting of messages.
- An asserting partner is set up within the federated network.

How to Configure a Resource Partner

Configuring a WS-Federation Resource Partner requires the following tasks:

1. Complete the SAML 1.x authentication scheme prerequisites.
2. Select the authentication scheme type and assign it a name.
3. Specify the namespace for users being authenticated with the SAML 1.x authentication scheme.
4. Select the single sign-on profile that this consumer supports (artifact or POST).

Configure a SAML authentication scheme for each Account Partner that is a federation partner and generates assertions. Bind each scheme to a realm, which includes the URLs of the target resources that users request. You can do this task on a per Account Partner basis or can create a single custom authentication scheme and single realm. Protect these resources with a CA SiteMinder® policy.

Tips:

- Certain parameter values at the Account Partner and Resource Partner must match for the configuration to work. A list of those parameters is available in Configuration Settings that Must Use the Same Values (see page 361).
- Verify that you are using the correct URLs for the Federation Web Services servlets. The URLs are listed in Federation Web Services URLs Used in SiteMinder Configuration (see page 367).

Optional Configuration Tasks for a Resource Partner

The optional tasks for configuring a Resource Partner include:

- Customize assertions using the Message Consumer Plug-in.
- Redirect failed authentication attempts.

Navigating Legacy Federation Dialogs

The Administrative UI provides two ways to navigate to the legacy federation configuration dialogs.

You can navigate in one of two ways:

- Following a wizard to configure a new legacy federation object.

  When you create an object, a page displays with a configuration wizard. Follow the steps in the configuration wizard to create the object.

- Selecting tabs to modify an existing legacy federation object.

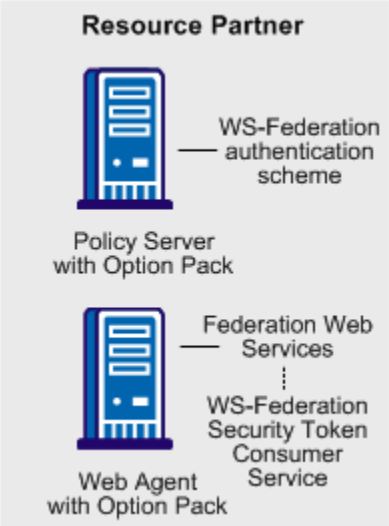
  When you modify an existing object, a page displays with a series of tabs. Modify the configuration from these tabs. These tabs are the same as the steps in the configuration wizard.

WS-Federation Authentication Scheme Overview

Any CA SiteMinder® site, or SPS Federation Gateway, can consume a <RequestSecurityTokenResponse> message and can use the assertion in the response to authenticate and authorize users. If sites in your federated network have user stores, you can use WS-Federation authentication.

The WS-Federation authentication scheme lets a Resource Partner authenticate a user. The authentication scheme enables cross-domain single sign-on by consuming a SAML assertion and establishing a CA SiteMinder® session. After the user is identified, the Resource Partner site can authorize the user for specific resources.

A site can be both a WS-Federation Resource Partner and Account Partner.



Note: The SPS federation gateway can replace the Web Agent and Web Agent Option Pack to provide the Federation Web Services application functions. For information about installing and configuring the SPS federation gateway, see the *Secure Proxy Server Administration Guide*.

The WS-Federation authentication scheme is configured at the Resource Partner-side Policy Server. The WS-Federation Security Token Consumer Service invokes the authentication scheme. The Security Token Consumer Service is a component of the Federation Web Services application and is installed on the Resource Partner-side Web Agent. This service obtains information from the WS-Federation authentication scheme at the Policy Server. FWS uses that information to extract the necessary information from the assertion to authenticate a user.

The SAML assertion becomes the user credentials to log in to the Policy Server at the Resource Partner site. The user is authenticated and authorized, and if authorization is successful, the user is redirected to the target resource.

Select the WS-Federation Authentication Scheme Type

The WS-Federation authentication scheme provides information about the Account Partner that generates the assertion for the Resource Partner. The authentication scheme specifies how the Resource Partner supports the authentication process.

After you configure an authentication scheme, associate the scheme with a realm that contains the resource you want to protect.

To configure a WS-Federation authentication scheme

1. Navigate to Infrastructure, Authentication, Authentication Schemes.

   The Create Authentication Scheme dialog opens.

2. From the Authentication Scheme Type drop-down list, select WS-Federation Template.

   The contents of the Authentication Scheme dialog change for the scheme.

   Note: Click Help for descriptions of settings and controls, including their respective requirements and limits.

After you select the authentication scheme template, you can configure the details of the authentication scheme. You access the rest of the configuration dialogs by clicking WS-Federation Configuration.

More Information:

How To Protect a Target Resource with a WS-Federation Authentication Scheme (see page 314)

Specify the General Information for the WS-Fed Auth Scheme

Identity the Resource Partner and the Account Partner in the General settings for the WS-Federation authentication schemes.

Follow these steps:

1. From the main authentication scheme page, click WS-Federation Configuration.

   If you are modifying an existing scheme, click Modify then WS-Federation Configuration.

   The detailed settings for the scheme display.

2. In the General settings, complete the required fields.

3. Verify that the Disable Signature Processing option is set appropriately for single sign-on.

   Important! For debugging purposes only, you can temporarily disable all signature processing (both signing and verification of signatures) by enabling the Disable Signature Processing option.

The general configuration is complete.

Locate User Records for Authentication

When you configure an authentication scheme, you define a way for the authentication scheme to look up a user in the local user store. After the correct user is located, the system generates a session for that user. Locating the user in the user store is the process of disambiguation. How CA SiteMinder® disambiguates a user depends on the configuration of the authentication scheme.

For successful disambiguation, the authentication scheme first determines a LoginID from the assertion. By default, the LoginID is extracted from the Name ID value in the assertion. You can also obtain the LoginID by specifying an Xpath query.

After the authentication scheme determines the LoginID, CA SiteMinder® checks if a search specification is configured for the authentication scheme. If no search specification is defined for the authentication scheme, the LoginID is passed to the Policy Server. The Policy Server uses the LoginID together with the user store search specification to locate the user. For example, imagine that the LoginID value is Username and the LDAP search specification is set to the uid attribute. The Policy Server uses the uid value (Username=uid) to search for the user.

If a search specification is configured for the authentication scheme, the LoginID is not passed to the Policy Server. Instead, the search specification is used to locate a user.

The disambiguation process involves two steps:

1. Obtain the LoginID by the default behavior or by using an Xpath query.
2. Locate the user in the user store by the default behavior or with a search specification.

Note: The use of Xpath and the search specification are optional.

Obtain a LoginID for a WS-Federation User

You can find the LoginID in two ways:

- Relying on the default behavior, where the LoginID is extracted from the NameID in the assertion. This option requires no configuration.
- Using an Xpath query to find the LoginID in place of the default behavior.

To specify an Xpath query

1. Navigate to the authentication scheme for the Resource Partner you are configuring.
2. Select WS-Federation Configuration, SAML Profiles. Click Modify first if you are modifying an existing scheme.

   The SAML Profiles dialog opens.
3. In the User Disambiguation section, enter an Xpath query that the authentication scheme uses to obtain a LoginID, then click OK.

   Xpath queries must not contain namespace prefixes. The following example is an invalid Xpath query:

   ```
   /saml:Response/saml:Assertion/saml:AuthenticationStatement/
   saml:Subject/saml:NameIdentifier/text()
   ```

   The valid Xpath query is:

   ```
   //Response/Assertion/AuthenticationStatement/Subject/
   ```

   NameIdentifier/text()

Use a Search Specification to Locate a WS-Federation User

You can use a search specification to locate the user in place of the default behavior, where the LoginID is passed to the Policy Server.

To locate a user with a search specification

1. Navigate to the authentication scheme for the Resource Partner you are configuring.
2. Select WS-Federation Configuration, SAML Profiles. Click Modify first if you are modifying an existing scheme.

   The SAML Profiles dialog opens.
3. In the User Disambiguation section, enter a search specification in the appropriate namespace field. The search specification defines the attribute that the authentication scheme uses to search a namespace. Use %s in the entry as a variable representing the LoginID.

   For example, the LoginID has a value of user1. If you specify Username=%s in the Search Specification field, the resulting string is Username=user1. This string is verified against the user store to find the correct record for authentication.
4. Click OK.

Configure WS-Federation Single Sign-on at the Resource Partner

You configure the WS-Federation single sign-on binding for authentication in the SSO section of the SAML Profiles page. You can also enforce single use assertion policy to prevent the replaying of a valid assertion in this section.

Part of the single sign-on configuration is defining the Redirect Mode setting. The Redirect Mode specifies how the Policy Server sends assertion attributes, if available, to the target application. You can send assertion attributes as HTTP Headers or HTTP cookies.

The HTTP headers and HTTP cookies have size restrictions that assertion attributes cannot exceed. The size restrictions are as follows:

- For HTTP headers, CA SiteMinder® can send an attribute in a header up to the web server size limit for a header. Only one assertion attribute per header is allowed. See the documentation for your web server to determine the header size limit.
- For HTTP cookies, CA SiteMinder® can send a cookie up to the size limit for a cookie. Each assertion attribute is sent as its own cookie. The cookie size limit is browser-specific, and that limit is for all attributes being passed to the application, not for each attribute. See the documentation for your web browser to determine the cookie size limit.

To configure WS-Federation single sign-on

1. Navigate to the authentication scheme for the Resource Partner you are configuring.
2. Select WS-Federation Configuration, SAML Profiles. Click Modify first if you are modifying an existing scheme.

   The SAML Profiles dialog opens.
3. Complete the fields in the SSO section.

   Click Help for the field descriptions.
4. Click Submit.

Implement WS-Federation Signout

Sign-out is the simultaneous termination of all user sessions for the browser that initiated the sign-out. Closing all user sessions prevents unauthorized users from gaining access to resources at the Resource Partner.

Sign-out does not necessarily end all sessions for a user. For example, a user with two browsers open can have two independent sessions. Only the session for the browser that initiates the sign-out is terminated at all federated sites for that session. The session in the other browser is still active.

The Policy Server performs sign-out using a signoutconfirmurl.jsp. This page resides on the Identity Provider system. An Identity Provider initiates a sign-out request on behalf of a user. The JSP sends the sign-out request to each site where the user signed on during a given browser session. The user is then signed out.

A user can initiate a sign-out request only at an Identity Provider. The request is triggered by clicking a link that points to the appropriate servlet. The sign-out confirmation page must be an unprotected resource at the Identity Provider site.

Note: The Policy Server only supports the WS-Federation Passive Request profile for sign-out.

Enable Signout

To configure WS-Federation signout

1. Navigate to the authentication scheme you want to modify.
2. Select WS-Federation Configuration, SAML Profiles. Click Modify first if you are modifying an existing scheme.

   The SAML Profiles dialog opens.
3. In the Signout section, select the Enable Signout check box.
4. Enter a value for the Signout URL. The URL must begin with https:// or http://.
5. Click OK.

Create a Custom WS-Federation Authentication Scheme

You can use a custom WS-Federation authentication scheme that is written with the CA SiteMinder® Authentication API instead of the existing WS-Federation authentication template.

The main authentication scheme page includes the Library field in the Scheme Setup section of the page. This field contains the name of the shared library that processes SAML artifact authentication. Do not change this value, unless you have a custom authentication scheme.

The default shared library for HTML Forms authentication is smauthhtml.

Customize Assertion Processing with the Message Consumer Plug-in

The message consumer plug-in is a Java program that implements the Message Consumer Plug–in. The plug-in lets you implement your own business logic for processing assertions, such as rejecting an assertion and returning a status code. This additional processing works together with the standard processing of an assertion.

Note: For more information about status codes for authentication and disambiguation, see the *CA SiteMinder® Programming Guide for Java*.

During authentication, CA SiteMinder® first tries to process the assertion by mapping a user to its local user store. If CA SiteMinder® cannot find the user, it calls the postDisambiguateUser method of the message consumer plug-in.

If the plug-in successfully finds the user, CA SiteMinder® proceeds to the second phase of authentication. If the plug-in cannot map the user to a local user store, the plug-in returns a UserNotFound error. The plug-in can optionally use the redirect URL feature. Without the consumer plug-in, the redirect URLs are based on the error that the SAML authentication scheme generates.

During the second phase of authentication, CA SiteMinder® calls the postAuthenticateUser method of the message consumer plug-in, if the plug-in is configured. If the method succeeds, CA SiteMinder® redirects the user to the requested resource. If the method fails, you can configure the plug-in to send the user to a failure page. The failure page can be one of the redirect URLs that you can specify with the authentication scheme configuration.

Additional information about the message consumer plug-in can be found as follows:

- Reference information (method signatures, parameters, return values, data types), and the constructor for UserContext class, are in the *Java Developer Reference*. Refer to the MessageConsumerPlugin interface.
- Overview and conceptual information about authentication and authorization APIs, see the *CA SiteMinder® Programming Guide for Java*.

To configure the plugin

1. Install the CA SiteMinder® SDK, if you have not done so already.
2. Implement the MessageconsumerPlugin.java interface, which is part of the CA SiteMinder® SDK.
3. Deploy your message consumer plug-in implementation class.
4. Enable the message consumer plug-in in the Administrative UI.

Implement the MessageConsumerPlugin Interface

Create a custom message consumer plug-in by implementing the MessageConsumerPlugin.java interface. The minimum requirements for the implementation class are listed in the following procedure.

Follow these steps:

1. Provide a public default constructor method that contains no parameters.
2. Provide code so that the implementation is stateless. Many threads must be able to use a single plug-in class.

3. Implement methods in the interface as your requirements demand.

   The MessageConsumerPlugin includes the following four methods:

   init()

   Performs any initialization procedures that the plug-in requires. CA SiteMinder® calls this method once for each plug-in instance, when the plug-in is loaded.

   release()

   Performs any rundown procedures that the plug-in requires. CA SiteMinder® calls this method once for each plug-in instance, when CA SiteMinder® is shutting down.

   postDisambiguateUser()

   Provides processing to disambiguate a user when the authentication scheme is unable to do so. Alternatively, this method can add data for new federation users to a user store. This method receives the decrypted assertion. The decrypted assertion is added to the properties map passed to plug-in under the key "_DecryptedAssertion".

   postAuthenticateUser()

   Provides additional code to determine the final outcome of assertion processing, regardless of whether the Policy Server processing is a success or failure.

CA SiteMinder® provides the following samples of the Message Consumer plug-in class:

MessageConsumerPluginSample.java in *installation_home*\sdk\samples\messageconsumerplugin

MessageConsumerSAML20.java in *installation_home*\sdk\samples\authextensionsaml20

Deploy a Message Consumer Plug-in

After you have coded your implementation class for the MessageConsumerPlugin interface, compile it and verify that CA SiteMinder® can find your executable file.

To deploy the Message Consumer Plugin:

1. Compile the MessageConsumerPlugin Java file. The file requires the following dependent libraries, which are installed with the Policy Server:

   installation_home\siteminder\bin\jars\SmJavaApi.jar

   An identical copy of SmJavaApi.jar is installed with CA SiteMinder® SDK. The file is in the directory *installation_home*\sdk\java\SmJavaApi.jar.

   You can use either of them at development time.

2. When a plug-in class is available, in a folder or a jar file, modify the -Djava.class.path value in the JVMOptions.txt file. This step enables the plug-in class to load with the modified classpath. Locate the JVMOptions.txt file in the directory *installation_home*\siteminder\config.

   Note: Do not modify the classpath for the existing xerces.jar, xalan.jar, or SmJavaApi.jar.

3. Restart the Policy Server to pick up the latest version of MessageConsumerPlugin. This step is necessary each time the plug-in Java file is recompiled.

4. Enable the plug-in.

Enable the Message Consumer Plug-in for WS-Federation

After writing a message consumer plug-in and compiling it, enable the plug-in by configuring settings in the Administrative UI. The UI settings tell CA SiteMinder® where to find the plug-in.

Do not configure the plug-in settings until you deploy the plug-in (see page 159).

To enable the message consumer plug-in

1. Log on to the Administrative UI

2. Navigate to the Authentication Scheme dialog for the appropriate WS-Federation scheme. In the General settings, go to the Advanced section and complete the following fields:

   Full Java Class Name

   Specify the Java class name for the plug-in, For example, a sample class included with the CA SiteMinder® SDK is:

   com.ca.messageconsumerplugin.MessageConsumerPluginSample

   Parameter

   Specify a string of parameters that are passed to the plug-in specified in the Full Java Class Name field.

   As an alternative to configuring the plug-in in the Administrative UI, use the Policy Management API (C or Perl) to set the IdpPluginClass and IdpPluginParameters.

3. Restart the Policy Server.

Redirect Users After Failed WS-Federation Authentication Attempts

For single sign-on processing, you can configure several optional redirect URLs if a user cannot be authenticated at the Resource Partner. The redirect URLs allow finer control over where a user is redirected. For example, when a user cannot be located in a user store, the Redirect URL for the User Not Found specification can redirect the user to the appropriate location.

Note: These URLs are not required.

If you do not configure redirect URLs, standard SiteMinder processing takes place. How SiteMinder handles a failed authentication depends on the configuration.

If a Resource Partner cannot authenticate a user during a single sign-on transaction, the Resource Partner can redirect that user to a customized URL for further processing.

You can configure several optional redirect URLs for failed authentication. If the assertion is not valid, the redirect URLs allow finer control over where a user is redirected. For example, if a user cannot be located in a user store, you can fill in a User Not Found redirect URL.

The Status Redirect URLs and Modes are in the Additional Configuration section of the authentication dialog. The redirect URLs are for specific status conditions:

- User is not found
- Single sign-on message is invalid
- User credentials are not accepted

If any of the conditions occur, redirect URLs can send the user to an application or a customized error page for further action.

Note: Configuring redirect URLs is not required.

To configure optional Redirect URLs

1. Navigate to the WS Federation authentication scheme you want to modify.
2. Select WS-Federation Configuration.
3. In the Advanced section, fill in a URL for one or more of the following fields:
   - Redirect URL for the User Not Found status
   - Redirect URL for the invalid SSO Message status
   - Redirect URL for the Unaccepted User Credential (SSO Message) status

   If enter a value for the Redirect URL for the Invalid SSO Message status, select a mode.

   Federation Web Services handles the errors by mapping the authentication reason into one of the configured redirect URLs. The user can be redirected to that URL to report the error.

Note: These redirect URLs can be used with the CA SiteMinder® message consumer plug-in for further assertion processing. If authentication fails, the plug-in can send the user to one of the redirect URLs you specify.

Supply SAML Attributes as HTTP Headers

An assertion response can include attributes in the assertion. These attributes can be supplied as HTTP header variables so a client application can use them for finer grained access control.

The benefits of including attributes in HTTP headers are as follows:

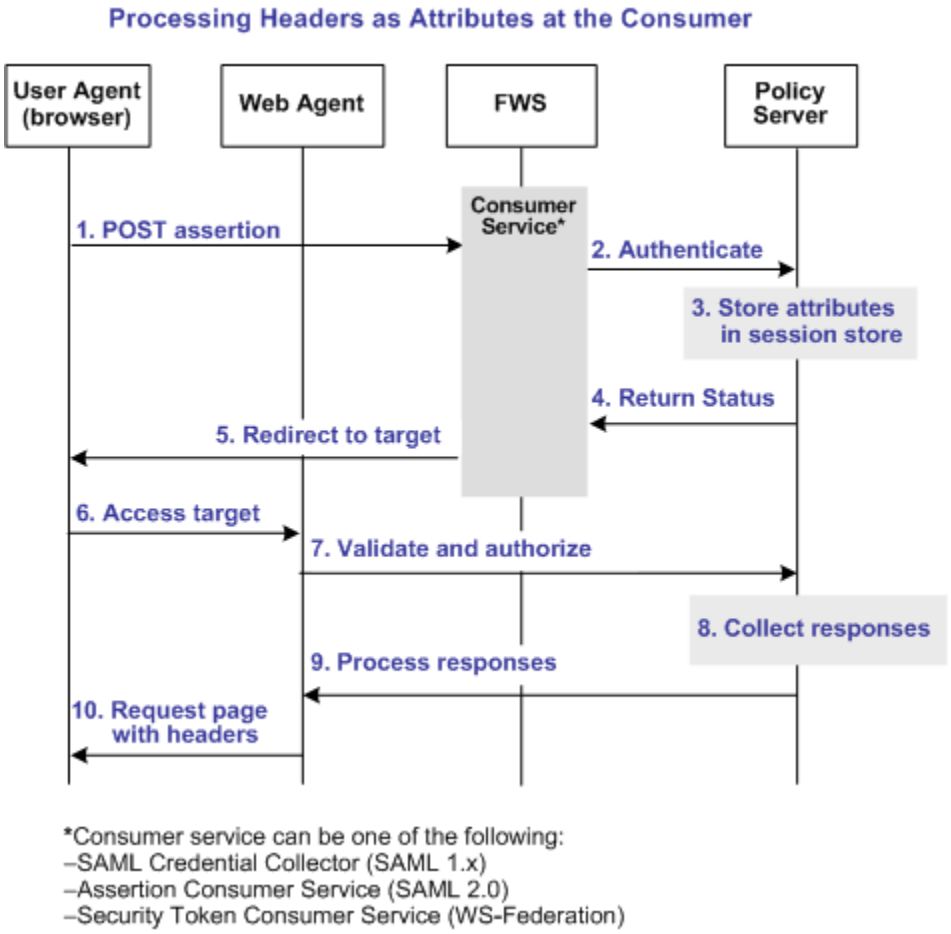
- HTTP headers are not persistent. They are present only within the request or response that contains them.
- HTTP headers, as supplied by the CA SiteMinder® Web Agent, are not visible in the browser, which reduces security concerns.

Note: The HTTP headers have size restrictions that the attributes cannot exceed. CA SiteMinder® can send an attribute in a header up to the web server size limit for a header. Only one assertion attribute per header is allowed. See the documentation for your web server to determine the header size limit.

Use Case for SAML Attributes As HTTP Headers

During authentication, a series of SAML attributes are extracted from an assertion and supplied as HTTP headers. During the authorization process, these headers are returned to the customer application.

The following flow diagram shows the sequence of events at runtime:



Note: The SPS federation gateway can replace the Web Agent and Web Agent Option Pack to provide the Federation Web Services application functions. In the flow diagram, the Web Agent block would be the embedded Web Agent in the SPS federation gateway. For information about installing and configuring the SPS federation gateway, see the *Secure Proxy Server Administration Guide*.

To process the attributes as HTTP headers, the sequence of events is as follows:

1. After the assertion is generated at the asserting party, it sends the assertion to the appropriate consumer service at the relying party. The delivery mechanism (POST or Artifact or WS-Fed) is irrelevant.

   Note: The consumer service can be the SAML credential collector (SAML 1.x), the Assertion Consumer Service (SAML 2.0), or Security Token Consumer Service (WS-Federation).

2. The consumer service calls its local Policy Server to use the configured authentication scheme to authenticate the user with the assertion.

3. If the authentication scheme redirect mode parameter is set to PersistAttributes, the Policy Server caches the attributes in the session store as session variables.
4. The result of the authentication is returned to the consumer service.
5. The consumer service redirects the browser to the protected target resource.
6. The browser tries to access the target resource.
7. The Web Agent calls the Policy Server to validate the user session and to verify that the user is authorized to access the target resource.
8. The Policy Server retrieves the attributes by a configured response.
9. The Policy Server processes the responses and sends the attributes to the Web Agent.
10. The Web Agent sets the HTTP headers as necessary.

Configuration Overview to Supply Attributes as HTTP Headers

Several configuration steps are required to retrieve the SAML attributes cached in the session store and provide them as HTTP headers.

Follow these steps:

1. Select PersistAttributes as the redirect mode for the SAML authentication scheme, which enables the SAML Attributes to be returned as HTTP headers.
2. Configure an authorization rule for the realm that contains the target resource.
3. Set PersistentRealm in the realm protecting the target resource.
4. Configure a response that uses the active response type for each SAML attribute to be supplied as a header.
5. Create a policy that binds the authorization rule and active response to implement the user of attributes as HTTP headers.

Set the Redirect Mode to Store SAML Attributes

After the relying party authenticates the user with the SAML assertion, the SAML attributes are written to the session store. The browser is then redirected to the target resource.

To redirect the browser with the attribute data

1. Log in to the Administrative UI.
2. Navigate to the configuration page of the SAML authentication scheme.

3. Set the Redirect Mode parameter to Persist Attributes. Locate the Redirect Mode field as follows:

   SAML 1.x

   The Redirect Mode is in the Scheme Setup section of the main configuration page.

   SAML 2.0

   Click SAML 2.0 Configuration, SSO. The Redirect Mode is in the SSO section of the page.

   WS-Federation

   Click WS-Federation Configuration, SAML Profiles. The Redirect Mode is in the SSO section of the page.

4. Click Submit to save your changes.

The redirect mode is now set to pass on the attribute data.

Create an Authorization Rule to Validate Users

For the realm containing the protected target resource, create a rule to retrieve the SAML attributes from the session store.

The rule is based on an authorization event (onAccessAccept). The user is already authenticated by the FWS application. The Web Agent cannot reauthenticate the user and then pass on the HTTP headers. The retrieval of the attributes occurs during the authorization stage.

To create an OnAccessAccept Rule for the realm

1. Log on to the Administrative UI.
2. Navigate to Policies, Domain, Realms.
3. Select the realm with the target resource.
4. Click Create in the Rules section.

   The Create Rule page appears.
5. Enter a name and optionally, a description.
6. Enter an asterisk (*) in the Resource field.
7. Select Authorization events and OnAccessAccept in the Action section.
8. Select Enabled in the Allow/Deny and Enable/Disable section.
9. Click OK to save the rule.

The authorization rule is now defined for the realm with the protected resource.

Configure a Response to Send Attributes as HTTP Headers

Configure a response that sends the SAML attributes as HTTP headers to the Web Agent. The Web Agent processes the response and makes the header variables available to the client application.

Follow these steps:

1. Log on to the Administrative UI.
2. Navigate to Policies, Domain, Domains.
3. Select the domain for the target resource and click Modify.
4. Select the Responses tab.
5. Click Create.

   The Response dialog opens.
6. Enter a name.
7. Confirm that the Agent type is a CA SiteMinder® Web Agent.
8. Click Create Response Attribute.

   The Response Attribute dialog opens.
9. Select WebAgent-HTTP-Header-Variable in the Attribute field.
10. Select Active Response for Attribute Kind.
11. Complete the fields as follows:

    Variable Name

    Specify the name that you want for the header variable. You assign this name.

    Library Name

    smfedattrresponse

    This value must be the entry for this field.

    Function Name

    getAttributeValue

    This value must be the entry for this field.

    Parameters

    Specify the name of the attribute as it appears in the assertion.

    An agreement between you and your federated partner determines the attributes that are in the assertion.
12. Click OK to save the attribute.

13. Repeat the procedure for each attribute that is to become an HTTP header variable. You can configure many attributes for a single response.

    You return to the Response tab. The attributes that you create are listed in the Attributes List section.

14. Click OK to save the response.

    You return to the Response tab.

15. Click Submit to save the domain.

The response sends the attributes on to the Web Agent to become HTTP headers.

Create a Policy to Implement Attributes as HTTP Headers

To implement the use of SAML attributes as HTTP headers, group together the authorization event rule and active response in a policy.

Follow these steps:

1. Log on to the Administrative UI.
2. Navigate to Policies, Domain, Domains.
3. Select the domain that contains the target resource and click Modify.
4. Select the Policy tab and click Create in the Policy section.

   The Create Policy dialog opens.

5. Enter a descriptive name in the Name field.
6. Select the users who are to have access to the protected resource in the Users tab.
7. Add the authorization rule that you created previously on the Rules tab.
8. Select the authorization rule and click Add Response.

   The Available Responses dialog opens.

9. Select the active response that you created previously and click OK.

   You return to the Rules tab. The response appears with the authentication rule.

10. Click Submit to save the policy.

The policy that enables SAML attributes to be used as HTTP headers is complete.

How To Protect a Target Resource with a WS-Federation Authentication Scheme

Protect target federation resources by configuring a CA SiteMinder® policy that uses the WS-Federation authentication scheme.

Follow these steps:

1. Create a realm that uses the WS-Federation authentication scheme. The realm is the collection of target resources.

   You can create a realm in the following ways:

   - Create a unique realm (see page 170) for each authentication scheme already configured.
   - Configure a single target realm (see page 170) that uses a custom authentication scheme to dispatch requests to the corresponding WS-Federation authentication schemes. Configuring one realm with a single target for all producers simplifies configuration of realms for authentication.

2. Configure an associated rule and optionally, a response.
3. Group the realm, rule, and response into a policy that protects the target resource.

Important! Each target URL in the realm is also identified in an unsolicited response URL. An unsolicited response is sent from the Account Partner to the Resource Partner, without an initial request from the Resource Partner. In this response is the target. At the Account Partner, an administrator includes this response in a link. The link redirects the user to the Resource Partner.

Configure a Unique Realm for Each Authentication Scheme

The procedure for configuring a unique realm for each SAML or WS-Federation authentication scheme follows the standard instructions for creating realms.

Follow these steps:

1. Navigate to Policy, Domain, Domains.

   The page to create domains displays.

2. Click Create Domain.
3. Enter a domain name.
4. Add the user directory to the domain. This directory is the one that contains the users requesting access to federated resources.

5. Select the Realm tab and create a realm.
   - In the Agent field, select the Web Agent protecting the web server where the target resources reside.
   - Select the appropriate authentication scheme in the Authentication Scheme field.
6. Create a rule for the realm.

   As part of the rule, select an action (Get, Post, or Put) that allows you to control processing when users authenticate.
7. Select the Policies tab and configure a policy that protects the target federation resource. Associate the realm that you previously created with this policy.

A policy with a unique realm now protects the federated resources.

Configure a Single Target Realm for All Authentication Schemes

To simplify configuration of realms for authentication schemes, create a single target realm for multiple sites generating assertions.

To do this task, set up the following components:

- A single custom authentication scheme

  This custom scheme forwards requests to the corresponding SAML or WS-Federation authentication schemes that you already configured for each asserting party.
- A single realm with one target URL

Create Authentication Schemes for the Single Target Realm

To define a custom authentication scheme for a single target realm, you must:

- Configure the authentication schemes.
- Define a parameter in the custom scheme that tells the Policy Server which authentication schemes to apply to resource requests.

First, verify that there are configured SAML or WS-Federation authentication schemes. If not, configure these schemes that the custom scheme can reference.

To create the authentication scheme

1. Navigate to Infrastructure, Authentication, Authentication Schemes.

   The Create Authentication Scheme page appears.

2. Create one or more authentication schemes according to the procedures for the protocol you are using.

3. Click OK to exit.

More information:

SAML 1.x Authentication Schemes (see page 148)
WS-Federation Authentication Scheme Overview (see page 297)
How to Configure a SAML 2.0 Authentication Scheme (see page 235)

Create the Custom Authentication Scheme

A single target realm relies on a specific custom authentication scheme to work properly.

To configure a custom authentication scheme for a single target realm

1. Navigate to Infrastructure, Authentication, Authentication Schemes.

   The Create Authentication Scheme page appears.

2. Complete the fields as follows:

   Name

   Enter a descriptive name for the custom authentication scheme, such as SAML Custom Auth Scheme.

3. In the Scheme Common Setup section, complete the following fields:

   Authentication Scheme Type

   Custom Template

   Protection Level

   Accept the default of set a new level.

4. In the Scheme Setup section, complete the following fields:

   Library

   smauthsinglefed

   Secret

   Leave this field blank.

   Confirm Secret

   Leave this field blank.

Parameter

Specify one of the following parameters:

- SCHEMESET=LIST; <*saml-scheme1*>;<*saml_scheme2*>

  Specifies the list of SAML authentication scheme names to use. If you configured an artifact scheme named artifact_producer1 and POST profile scheme named samlpost_producer2, you enter these schemes. For example:

  SCHEMESET=LIST;artifact_producer1;samlpost_producer2

- SCHEMESET=SAML_ALL;

  Specifies all the configured schemes. The custom authentication scheme enumerates all the SAML authentication schemes and finds the one with the correct Provider Source ID for the request.

- SCHEMESET=SAML_POST;

  Specifies all the SAML POST Profile schemes that you have configured. The custom authentication scheme enumerates the POST Profile schemes and finds the one with the correct Provider Source ID for the request.

- SCHEMESET=SAML_ART;

  Specifies all the SAML artifact schemes that you have configured. The custom authentication scheme enumerates the artifact schemes and finds the one with the correct Provider Source ID for the request.

- SCHEMESET=WSFED_PASSIVE;

  Specifies all the WS-Federation authentication schemes to find the one with the correct Account Partner ID.

Enable this scheme for CA SiteMinder® Administrators

Leave unchecked.

5. Click Submit.

The custom authentication scheme is complete.

Configure the Single Target Realm

After you configure the authentication schemes and associate them with a custom scheme, configure a single target realm for federation resources.

Follow these steps:

1. Navigate to Policies, Domain, Domains.
2. Modify the policy domain for the single target realm.
3. Select the Realms tab and click Create.

   The Create Realm dialog opens.

4. Enter the following values to create the single target realm:

   Name

   Enter a name for this single target realm.

5. Complete the following field in the Resource option:

   Agent

   Select the Web Agent protecting the web server with the target resources.

   Resource Filter

   Specify the location of the target resources. The location is where any user requesting a federated resource gets redirected.

   For example, /FederatedResources.

6. Select the Protected option in the Default Resource Protection section.
7. Select the previously configured custom authentication scheme in the Authentication Scheme field.

   For example, if the custom scheme was named Fed Custom Scheme, you would select this scheme.

8. Click OK.

The single target realm task is complete.

Configure the Rule for the Single Target Realm

After you configure the single target realm, configure a rule to protect the resources.

1. Navigate to the Modify page for the single target Realm.
2. Click Create in the Rules section.

   The Create Rule page appears.

3. Enter values for the fields on the rules page.
4. Click OK.

The single target realm configuration includes the new rule.

Create a Policy Using the Single Target Realm

Create a policy that references the single target realm. Remember that the single target realm uses the custom authentication scheme that directs requests to the appropriate SAML authentication scheme.

Note: This procedure assumes that you have already configured the domain, custom authentication scheme, single target realm and associated rule.

Follow these steps:

1. Navigate to the previously configured domain.
2. Select the Policies tab and click create.

   The Create Policy page opens.
3. Enter a name and a description of the policy in the General section.
4. Add users to the policy from the Users section.
5. Add the rule that you created for the single target realm from the Rules tab.

   The remaining tabs are optional.
6. Click OK.
7. Click Submit.

The policy task is complete. When a request triggers this policy, it relies on the single realm and associated authentication schemes to authenticate the user.

Chapter 18: Configure SAML 2.0 Affiliations

This section contains the following topics:

Affiliation Overview

A SAML affiliation is a group of SAML entities that share a name identifier for a single principal.

Service Providers and Identity Providers can belong to an affiliation. However, a single entity can belong to only one affiliation. Service Providers share the Name ID definition across the affiliation. Identity Providers share the user disambiguation properties across the affiliation.

Affiliations reduce the configuration that is required at each Service Provider. Additionally, using one name ID for a principal saves storage space at the Identity Provider.

Affiliations offer the following functions:

- Single sign-on
- Single logout

Note: Configuring affiliations is optional.

Affiliations for Single Sign-On

In a single sign-on use case, the Service Provider sends a request for an assertion to an Identity Provider. The AuthnRequest contains an attribute that specifies an affiliation identifier.

When the Identity Provider receives the request, it takes the following actions:

- Verifies that the Service Provider is a member of the affiliation identified in the AuthnRequest.
- Generates the assertion with the Name ID that is shared by the affiliation.
- Returns this assertion to the Service Provider.

Upon receiving the assertion, authentication takes place at the Service Provider.

Affiliations for Single Logout

When a Service Provider generates a logout request, it verifies whether the Identity Provider is a member of an affiliation. The Service Provider includes an attribute in the request, which it sets to the affiliation ID. The Identity Provider receives the request and verifies that the Service Provider belongs to the affiliation identified in the attribute.

The Identity Provider obtains the affiliation Name ID from the session store of the session store. When the Identity Provider issues logout request messages to all session participants, it includes the affiliation Name ID for the members of the affiliation.

Configure SAML 2.0 Affiliations

A SAML affiliation lets you add a SAML entity to a group so it can share a name identifier for a single principal. You can configure affiliations at either partner in a federated network.

For an Identity Provider, assign a name ID associated with an affiliation. The shared Name ID properties apply to all the Service Providers that belong to the affiliation.

For the Service Provider, the affiliation provides the user disambiguation process for authentication. When the Service Provider receives an assertion, it extracts the user identity information from the assertion. Based on the user disambiguation settings, the Service Provider compares the identity information against a local user directory to find the proper user record.

Follow these steps:

1. Navigate to Federation, Legacy Federation, SAML Affiliations.

   The Create SAML Affiliation page appears.

2. Complete the necessary fields. Note the following information:
   - For an Identity Provider, the Users settings have no function. Disregard these settings
   - For a Service Provider, the Name IDs settings have no function. Disregard these settings.
3. Click Submit.

A list of Service Providers that are members of the affiliation are displayed in the SAML Service Providers Associations section of the affiliate dialog. This list of Service Providers is a read-only list. To edit this list, modify the Service Provider object.

A list of SAML 2.0 authentication schemes that use an affiliation for user disambiguation is displayed in the SAML Authentication Scheme Associations section. This list of authentication schemes is a read-only list. To edit this list, modify the particular scheme.

Assign Affiliations at the Identity Provider

For the Identity Provider, the affiliation provides the Name ID in an assertion. Additionally, the Identity Provider includes an affiliation ID in the assertion. Select an affiliation when you configure a Service Provider object.

At runtime, the Identity Provider uses the NameID for the affiliation and disregards the Name ID configuration that is defined for the Service Provider object.

Follow these steps:

1. Navigate to the Service Provider object you want to modify.
2. Go to the Name IDs page.
3. Select a SAML affiliation from the pull-down list.

   The affiliation must already be configured to be in the list.

Assign Affiliations at the Service Provider

For the Service Provider, an affiliation determines user information. Select an affiliation when you configure an authentication scheme at the Service Provider.

At runtime, the Service Provider relies on the user configuration from the affiliation. It disregards the user configuration in the authentication scheme.

Follow these steps:

1. Navigate to the SAML 2.0 authentication scheme you want to modify.
2. Go to the General page.
3. In the User Disambiguation section, select a SAML affiliation from the pull-down list.

   The affiliation must already be configured to be in the list.

Chapter 19: Authorize Users with Attributes from an Assertion Query

This section contains the following topics:

Perform Authorizations with an Attribute Authority

The Policy Server authorizes a user with the following types of information:

- Users that are specified in the policy configuration
- Policy expressions
- Active policies
- IP address restrictions
- Time restrictions

The Policy Server also authorizes a user with user attributes that a SAML 2.0 Attribute Authority provides. When a user requests access to a protected resource, the Policy Server, as the authorizing entity, can request more user attributes. The Policy Server evaluates these attributes before granting access to the resource.

The SAML 2.0 Assertion Query/Request profile employs two entities, a **SAML Attribute Authority** and a **SAML Requester**.

SAML Attribute Authority

The SAML Attribute Authority relies on an Attribute Service to process a query message and add attributes to an assertion. These assertions contain user attributes that a SAML Requester uses to authorize access to protected resources. The Attribute Service is part of the Federation Web Services application.

When an entity makes a request to an Attribute Authority, the message contains the user attributes that the requester wants to retrieve. The message also contains the Name ID and the Issuer of the request. The Attribute Service uses the NameID to disambiguate the user so it knows what values to return for the requested attributes. The Attribute Service returns a response message that includes an attribute assertion that is wrapped in a SOAP message. This response includes the user attributes.

Note: The user does not need to be authenticated at the Attribute Authority. Also, there is no need for a single sign-on relationship between the Authority and the Requester.

SAML Requester

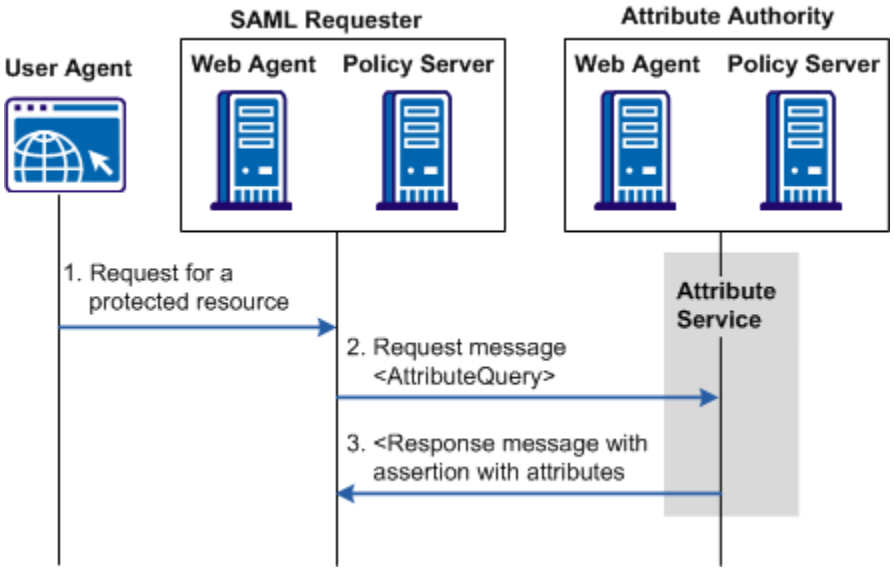
The SAML Requester is a SAML entity that uses the SAML 2.0 Assertion Query/Request profile to request attributes for a user. For CA SiteMinder®, the SAML Requester is not a specific service, but a group of Policy Server features that can produce and process <AttributeQuery> messages. The Requester asks for the user attributes from the Attribute Authority because the protected target resource always resides at the SAML requester. The Requester resolves these attributes into variables that a policy expression uses.

Note: In a CA SiteMinder® federated environment, the SAML Attribute Authority is the Identity Provider and the SAML Requester is the Service Provider. However, this condition does not have to be the case.

To evaluate an authorization request that is based on SAML 2.0 user attributes, add an attribute type named **federation attribute variable** to a policy expression. The policy protecting the target resource uses this variable. Based on the policy variable, the SAML Requester sends a query message to the Attribute Authority. This query message contains the Name ID for the SAML entity for which the attributes are being requested. The SAML Attribute Authority returns a response message containing assertions with the attribute statements.

A user must have a session at the SAML Requester; however, the user does not have to log in or authenticate at the Attribute Authority.

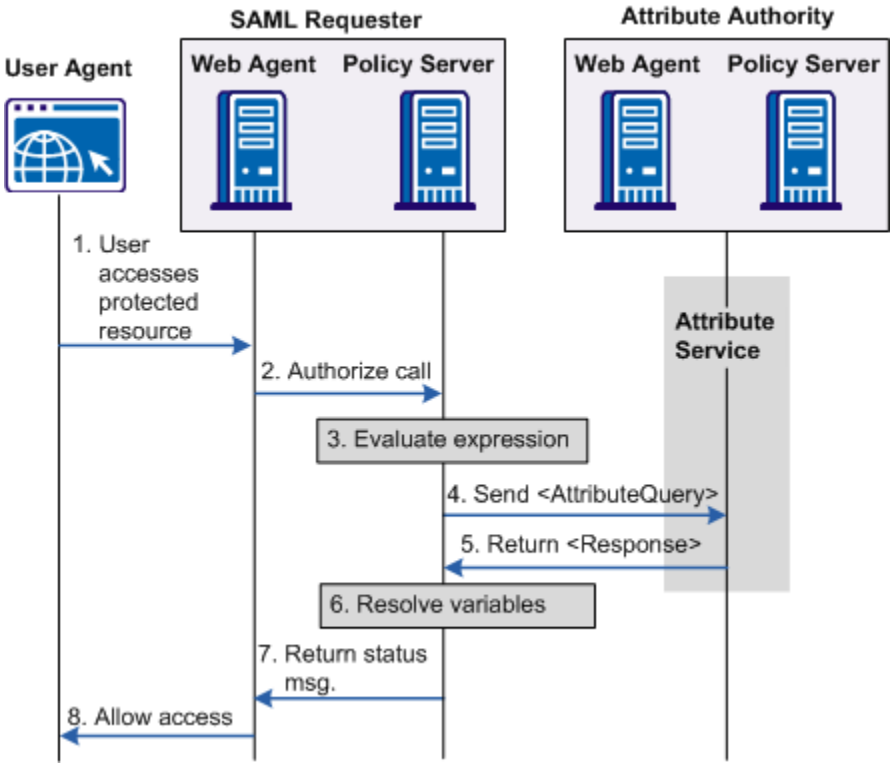
The following figure shows how an attribute query is processed.



Note: The SPS federation gateway can replace the Web Agent and Web Agent Option Pack to provide the Federation Web Services application functions. For information about installing and configuring the SPS federation gateway, see the *Secure Proxy Server Administration Guide*.

Flow Diagram for Authorizing a User with User Attributes

The following flow diagram shows the authorization process with an Attribute Authority.



Note: The SPS federation gateway can replace the Web Agent and Web Agent Option Pack to provide the Federation Web Services application functions. For information about installing and configuring the SPS federation gateway, see the *Secure Proxy Server Administration Guide*.

The sequence of a user attribute request is as follows:

1. A user accesses a protected resource. The user can log in locally or can be authenticated through a SAML assertion.
2. The Web Agent at the SAML Requester calls the local Policy Server determine whether the user is authorized to access the resource. The policy that protects the resource uses a policy expression for authorization with a federated attribute variable.
3. The Policy Server tries to resolve these variables but cannot. The Policy Server looks up the user in the local user store to obtain the NameID of the user.
4. An attribute query is sent to the AttributeService URL at the Attribute Authority. The AttributeQuery contains the users NameID and the requested attributes.

5. The Attribute Authority returns a SAML response containing an assertion with the requested attributes.
6. The SAML Requester completes the resolution of variables and then evaluates the policy expression.
7. An authorization status message is returned to the Web Agent.
8. Depending on the authorization status, the Web Agent allows or denies access to the requested resource.

How to Configure an Attribute Authority and a SAML Requester

In a CA SiteMinder® context, the Attribute Authority is the Identity Provider.

To configure CA SiteMinder® to act as a SAML Attribute Authority

1. Define a search specification for locating a user. Enter the NameID into the search specification.
2. Configure the back channel across which the Authority sends the response to a query.
3. Define the attributes that are returned in response to a query.
4. Grant users access to the attribute authority service.

In a CA SiteMinder® context, the SAML Requester is the Service Provider.

To configure CA SiteMinder® as a SAML Requester

1. Enable the attribute query functionality.
2. Configure the back channel across which the Requester receives the response from the Authority.
3. Define the list of attributes requested in the attribute query.
4. Configure the federation attribute variables.
5. Configure the NameID for inclusion in the attribute query.

Set up the Attribute Authority

In a CA SiteMinder® context, the Attribute Authority is the Identity Provider with the Attribute Authority service enabled.

Note: You do not need to configure other Identity Provider features, such as single sign-on to have the Identity Provider act as an Attribute Authority.

To configure an Attribute Authority

1. Log on to the Administrative UI.
2. Navigate to the Service Provider object that represents the SAML Requester. The SAML Requester requests the user attributes.
3. Select Modify.

   The SAML Service Provider page opens.
4. Select the Attributes tab.
5. In the Attribute Svc section, select Enable. This check box enables the Attribute Authority feature.
6. (Optional) Modify the value of the Validity Duration. You can accept the default of 60 seconds.

   Modify this setting only if you want the assertion to be valid for longer than 60 seconds.

   Note: Click Help for the field descriptions.
7. (Optional) Configure one or both of the signing settings. Neither setting is required.

   Require Signed Attribute Query

   Select this option if you want to the Attribute Authority to accept only signed queries from the SAML Requester.

   Signing Options

   Select one of the options to sign the attribute assertion, the SAML response, both, or neither when they are returned to the SAML Requester.
8. In the User Lookup section, specify a search specification for the namespace you want to use.

   Enter a namespace attribute that the authentication scheme uses as a search string.

   Use **%s** in the entry as the variable that represents the NameID. For example, the NameID has a value of user1. If you specify Username=%s in the Search Specification field, the resulting string is Username=user1. This string is verified against the user store to find the correct record for authentication.
9. In the Backchannel section, complete the following fields:
   - Password
   - Confirm Password

   If you configured SAML 2.0 artifact authentication, you have already configured a password for the back channel. This password can be used for both SSO and the Attribute Authority Service.
10. Click Submit to save your changes.
11. Go to Configure the Attributes at the Attribute Authority (see page 331).

Configure Attributes at the Attribute Authority

Indicate whether the attribute you are configuring is part of a single sign-on request, or an attribute query request. The Retrieval Method field in the SAML Service Provider Attribute dialog determines the attributes function.

To use the same attribute for both services, create two attribute statements that use the same Attribute name and variable. One attribute uses SSO as the retrieval method and one uses Attribute Services as the retrieval method.

To configure an attribute

1. Configure Attributes for SSO Assertions (see page 219).

   The configuration process for configuring attributes at the Attribute Authority is the same for configuring attributes for single sign-on assertions.

2. Navigate to the Attributes dialog for the Service Provider object that represents the SAML Requester.

3. From the Attributes dialog, select Add in the Attribute section.

   The Add Attribute page displays.

4. Select Attribute Service for the Retrieval Method field in the Attribute Setup section of the page.

   If an attribute query requests this attribute, selecting Attribute Service as the Retrieval Method marks the attribute for inclusion in the attribute assertion.

Grant Relying Partners Access to the Attribute Authority Service

For the Attribute Authority Service to respond to requests, you must give the relying partners access to the service. This is a two-step process:

1. Add a Web Agent to the Federation Agent Group (see page 130)
2. Add Relying Partners to the Polciy for the Attribute Authority Service. (see page 332)

Add a Web Agent to the Federation Agent Group

Add the Web Agent that protects the FWS application to the Agent group FederationWebServicesAgentGroup.

- For ServletExec, this Agent is on the web server where the Web Agent Option Pack is installed.
- For an application server, such as WebLogic or JBOSS, this Web Agent is installed where the application server proxy is installed. The Web Agent Option Pack can be on a different system.

Follow these steps:

1. Log in to the Administrative UI.
2. Click Infrastructure, Agent, Agents.
3. Click Create Agent.
4. Specify the name of the Web Agent in your deployment. Click Submit.
5. Click Infrastructure, Agent, Agent Groups.
6. Select the FederationWebServicesAgentGroup entry.
7. Click Add/Remove and the Agent Group Members dialog opens.
8. Move the web agent from the Available Members list to the Selected Members list.
9. Click OK to return to the Agent Groups dialog.
10. Click Submit then click Close to return to the main page.

Add Relying Partners to the Policy for the Attribute Authority Service

If you are implementing authorizations with an Attribute Authority, the relying party in the partnership needs permission to access the attribute authority service. CA SiteMinder® protects the SAML 2.0 attribute authority with a policy.

When you install the Policy Server, the FederationWebServicesDomain is installed by default. This domain includes the SAML2FWSAttributeServicePolicy for the attribute service.

Grant access for the attribute service policy to any relevant relying partners.

Follow these steps:

1. In the Administrative UI, navigate to Policies, Domain, Domain Policies.

   A list of domain policies displays.

2. Select the SAML2FWSAttributeServicePolicy.

   The Domain Policies page opens.

3. Click Modify to change the policy.
4. Select the Users tab.
5. In the dialog for the SAML2FederationCustomUserStore user directory, click Add Members.

   The User/Groups page opens.

   The affiliate domain that you previously configured is listed in the Users/Groups dialog. For example, if the affiliate domain is named fedpartners, the entry is **affiliate:fedpartners**.

6. Select the check box next to the affiliate domain with the partners that require access to the service. Click OK.

   You return to the User Directories list.

7. Click Submit.

   You return to the policies list.

The necessary relying partners now have access to the attribute authority service.

How to Set up a SAML Requester to Generate Attribute Queries

For a CA SiteMinder® Service Provider to act as a SAML Requester, configure a SAML 2.0 authentication scheme so that an attribute query can be generated. Complete this configuration at the Service Provider site.

Follow these steps:

1. Log on to the Administrative UI
2. Navigate to a SAML 2.0 authentication scheme configuration.
3. Enable attribute queries and specify attributes (see page 333).
4. Configure a Name ID for the attribute query (see page 334).
5. Configure the back channel for the attribute query. (see page 334)
6. Configure a federation attribute variable. (see page 335)
7. Create a policy expression with the federation attribute variable. (see page 336)

Each step is detailed in the following sections.

Enable Attribute Queries and Specify Attributes

For a SAML requester to generate attribute queries, enable the attribute query functionality.

Follow these steps:

1. Log on to the Administrative UI.
2. Access the authentication configuration for the SAML 2.0 authentication scheme.
3. Select the Attributes tab.
4. In the Attribute Query section, select Enabled.

5. (Optional) Select the following check boxes:
   - Sign Attribute Query
   - Require Signed Assertions
   - Get All Attributes
6. Enter a value for the Attribute Service field.
7. In the Attributes section of the page, click Add.

   The Add Attributes page opens.
8. Enter values for the fields on the page.
9. Click OK to save your changes.

   You return to the Attributes page.
10. Configure the NameID (see page 334). This NameID is included in the attribute query for use by the Attribute Authority.

Configure the NameID for the Attribute Query

A query message sent to the Attribute Authority includes the Name ID of the user whose attributes it is requesting. The Name ID configuration specifies how the SAML Requester obtains the Name ID. The requester then places the Name ID in the attribute query.

To specify a Name ID

1. Navigate to the Attributes dialog for the SAML Requester authentication scheme.
2. In the Name IDs section of the page, define the following settings:
   - Name ID Format
   - Name ID Type
   - Name ID Fields

   Click Help for the field descriptions.
3. Click OK to save your changes.
4. If the back channel is not already configured, configure it.

Configure the Backchannel for the Attribute Query

The attribute query is sent across a secure back channel to the Attribute Authority.

Only one back channel is available between the Service Provider and the Identity Provider. Therefore, the back channel configuration for the attribute query is the same back channel configuration that is used for the SAML artifact profile.

To configure the back channel

1. Navigate to the authentication scheme page for the SAML requester.
2. Click the Encryption & Signing tab.
3. In the Backchannel section, complete the following fields:
   - Authentication
   - SP Name
   - Password
   - Confirm Password

   Click Help for the field descriptions.
4. Click OK.

Create a Federation Attribute Variable

To use a federation attribute variable in a policy expression, first create the attribute variable.

To define a federation attribute variable

1. Navigate to Policies, Domain, Variables.

   The Variables dialog opens.
2. Select Create Variable.

   The first step of the configuration wizard displays the Domain section.
3. Select the federation policy domain where you plan to add the variable and click Next.
4. In the Define Variables step, complete the two fields in the General section.

   Name

   Identifies the variable.

   Variable Type

   Federation Attribute
5. Complete the fields in the Definition section.

   Click Help for the field descriptions.

6. Click Finish to save the variable.
7. Add this variable to a policy expression. (see page 336) The policy that protects a federated resource uses the policy expression.

Note: A policy expression can use multiple federation attribute variables; each variable is tied to a SAML 2.0 authentication scheme. Therefore, a single expression can result in many attribute requests sent to many Attribute Authorities.

Create a Policy Expression with the Federation Attribute Variable

To use a federation attribute variable as part of the authorization process, add the attribute variable to a policy expression. Associate this policy expression with the policy protecting the target resource at the SAML requester.

For information on creating a policy expression, see the Policies chapter in the *Policy Server Configuration Guide*.

Chapter 20: Use SAML 2.0 Provider Metadata To Simplify Configuration

This section contains the following topics:

Metadata Tools for SAML 2.0

The Policy Server provides a metadata tool to import and export SAML 2.0 metadata programmatically. Metadata lets you efficiently exchange federation configurations between a site that uses CA SiteMinder® and a partner that uses a third party or CA SiteMinder®. Programmatic use of SAML 2.0 metadata can limit how much configuration that you perform.

The two command line utilities that make up the metadata tools are smfedexport and smfedimport.

Exporting metadata involves the following types of input:

- User input
- Access to the certificate data store for including KeyInfo into the metadata
- Access to the certificate data store for signing
- Access to the policy store to reference similar metadata that can be used as a template.

Importing metadata involves:

- User input.
- Access to the policy store.
- Access to the certificate data store for verifying signatures, if certificates are configured.
- Parsing the XML metadata in the metadata document.
- Storing the relevant metadata in the policy store.
- Storing the PKI information from the metadata in the certificate data store.

Export Metadata Tool

You can use the export tool in the following situations:

- Create an Identity Provider metadata file for use by Service Providers.

  Use the tool to produce a metadata file containing information about profiles that the Identity Provider supports. This XML output that the export tool generates describes the Identity Provider. Sites acting as Service Providers can import this metadata file to establish a relationship with the Identity Provider.

- Create an Identity Provider metadata file from an existing Service Provider.

  A CA SiteMinder® Identity Provider generates a metadata file from an existing Service Provider object. The use of the Service Provider object reduces the amount of required data that a user must configure. Many of the settings for the Identity Provider metadata file can be derived from the existing Service Provider. Also,CA SiteMinder® provides the default names of the servlets.

  To use the metadata file, the existing relationship between the Identity Provider and the Service Provider is similar to the relationship you are establishing.

  The SSO and SLO servlet URLs are the default servlet names that are prepended with the IP address and port of the Federation Web Services application.

  The servlet names are:

  - http://*idp_server:port*/affwebservices/public/saml2sso
  - http://*idp_server:port*/affwebservices/public/saml2slo

  idp_server:port

  Identifies the web server and port hosting the Web Agent Option Pack or SPS federation gateway.

- Create a Service Provider metadata file for Use by Identity Providers.

  A CA SiteMinder® Service Provider can facilitate federation with sites acting as Identity Providers by producing a metadata file containing information about the profiles it supports. An Identity Provider can import the metadata file to establish a relationship with the Service Provider.

- Create a Service Provider metadata file from an existing SAML 2.0 Authentication Scheme.

  A CA SiteMinder® Service Provider generates a metadata file from an existing SAML 2.0 Authentication Scheme object. The use of the Service Provider object reduces the amount of required data that a user must configure. Many of the settings for the SP metadata file can be derived from the existing SAML 2.0 authentication scheme. CA SiteMinder® provides the default names of the servlets.

To use the metadata file, the existing relationship between the Service Provider and the Identity Provider must be similar to the relationship you are establishing. The SSO and SLO servlet URLs are the default servlet names that are prepended with the IP address and port of the Federation Web Services application.

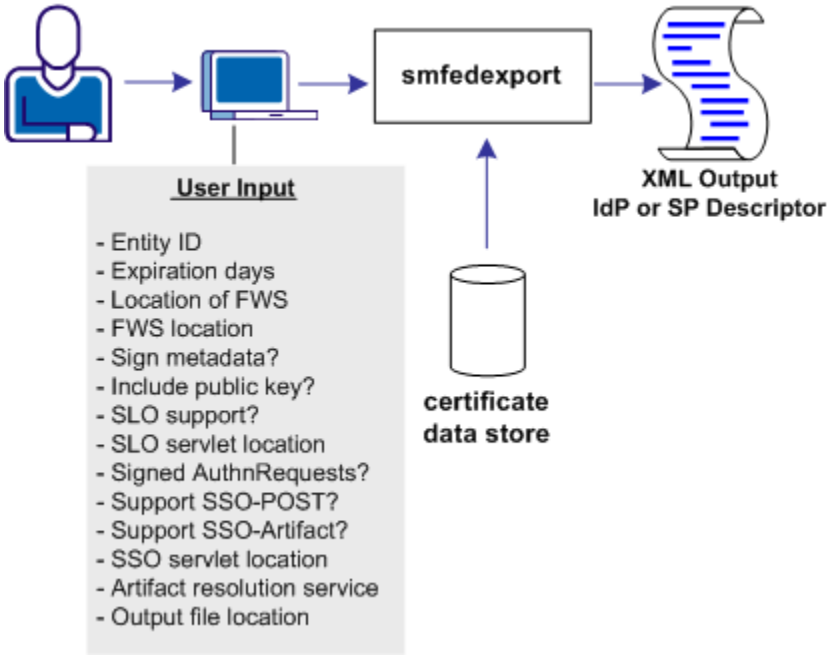
The servlets are:

- http://*idp_server:port*/affwebservices/public/saml2sso
- http://*idp_server:port*/affwebservices/public/saml2slo

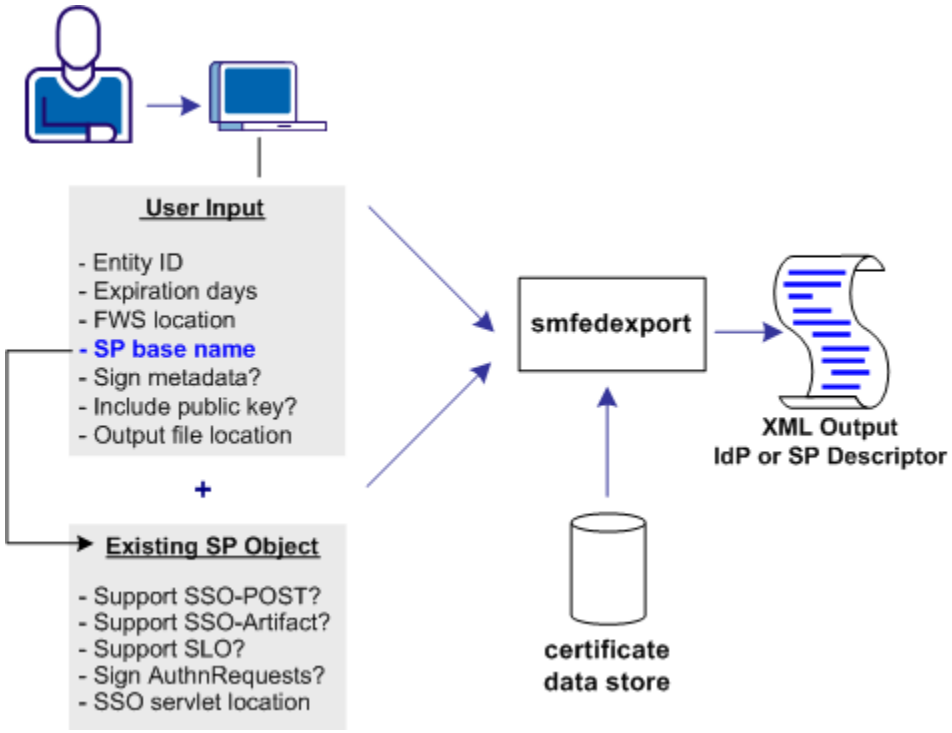
idp_server:port

Identifies the web server and port hosting the Web Agent Option Pack or SPS federation gateway.

The following illustration shows a metadata file that is generated only from user input.



The following illustration shows a metadata file that is generated from a combination of user input and data from an existing Service Provider object.



Run the smfedexport Tool

The smfedexport tool lets you export SAML 2.0 metadata to an XML file.

If you enter smfedexport without any command arguments, all the command arguments and their usage are displayed.

To run the smfedexport tool

1. At the system where you installed the Policy Server, open a command window.
2. Enter the smfedexport command using the syntax for the task you want to complete:

   Note: Command arguments enclosed in square brackets [] are optional.

   To export a SAML 2.0 Identity Provider metadata file:

   ```
   smfedexport -type saml2idp [-entityid <entityid>] [-expiredays <num>]
   [-fwsurl <FWS Location> [-spbase <spname>] -username <SiteMinder Admin Name>
   -password <SiteMinder Admin Password>]][-sign][-pubkey]
   [-slo <SLO Service Location> -slobinding <REDIR>] [-reqsignauthr]
   [-sso <SSO Service Location> -ssobinding <REDIR|SOAP>]
   [-ars <Artifact Resolution Service Location>][-output <file>]
   ```

To export a SAML 2.0 Service Provider metadata file:

```
smfedexport -type saml2sp [-entityid <entityid>] [-expiredays <num>]
[-fwsurl <FWS Location> [-schemebase <Auth Scheme name>
-username <SiteMinder Admin Name> -password <SiteMinder Admin Password>]]
[-sign][-pubkey][-slo <SLO Service Location> -slobinding <REDIR>]
[-signauthr][-acs <Assertion Consumer Service> -acsbinding <ART|POST|PAOS>
-acsindex <num>][-acsisdef]][-output <file>]
```

To sign an existing Metadata document:

```
smfedexport -type (saml2sp|saml2idp) -sign -input <file> -output <file>
```

After you run the tool, an XML file will be produced. If the -type option is set to saml2idp, the default output file name is IDPSSODescriptor.xml. If the -type option is set to saml2sp, the default output file name is SPSSODescriptor.xml.

After smfedexport processes the initial command options, the tool prompts you for additional data that is related to the type of export file the tool is generating. Any optional arguments that you do not enter use default values.

Note: If you are creating an IdP metadata file, you must have at least one single sign-on service defined in the smfedexport command. If you are creating an SP metadata file, you must have at least one assertion consumer service defined in the smfedexport command.

Command Options for smfedexport

The smfedexport command line options are listed in the table that follows:

Option	Description	Values
-acs	Assertion Consumer Service URL	URL
-acsindex	Assertion Consumer Service index value	integer
-acsisdef	Makes the immediately preceding Assertion Consumer Service the default.	none
-acsbinding	SAML protocol binding for the Assertion Consumer Service.	■ ART (artifact) ■ POST (POST) ■ PAOS (Reverse SOAP - ECP)
-ars	Artifact Resolution Service	URL

Option	Description	Values
-decryptionkeyalias	Tells the Policy Server to include the certificate (public key) in the metadata. This certificate encrypts the metadata. The SP decrypts the metadata using the corresponding private key.	alias name
-entityid	Represents the ID of the SP or IDP whose metadata you are exporting.	URI
-expiredays	Days until the metadata document is no longer valid.	integer; 0 is the default A value of 0 indicates that the metadata document has no expiration. No "validUntil" elements being generated in the exported XML.
-fwsurl	URL pointing to the FWS application.	URL in the form http://*host:port*
-input	Full path to an existing XML file	string, no default
-output	Full path to an output XML file	Default values: IDPSSODescriptor.xml SPSSODescriptor.xml
-password	SiteMinder Administrator name Requires the -username option	string, no default
-pubkey	Tells the Policy Server to include the certificate (public key) in the metadata. The partner site uses the certificate for signature encryption and verification. This setting is optional because the metadata does not have to be signed.	true, if present false otherwise
-reqsignauthr	Require signed AuthnRequests	true, if present false otherwise

Option	Description	Values
-schemebase	Points to an existing Service Provider. The settings for the profiles/bindings are taken from this provider. Requires the following options: -fwsurl -username -password	authentication scheme name
-spbase	Points to an existing Service Provider. The settings for the profiles/bindings are taken from this provider. Requires the following options: -fwsurl -username -password	Service Provider Name
-sign	Indicates whether the Policy Server signs the metadata. This setting is optional.	true, if present false, otherwise
-sigalg	Designates the signature hashing algorithm CA SiteMinder® uses to for signing assertions and assertion responses, single logout requests and responses.	rsawithsha1 rsawithsha256
-signauthr	Indicates whether the SP signs AuthnRequests	true, if present false, otherwise
-signingcertalias	Specifies the alias for the key/certificate pair that signs the metadata. Store the pair in the certificate data store. This setting is an alternative to the default alias, defaultenterpriseprivatekey. If you do not enter a value for this option, the Policy Server uses the defaultenterpriseprivatekey alias to sign the metadata.	alias name
-slo	Single Logout Service URL	URL
-slobinding	HTTP binding that is used for single logout. HTTP Redirect binding is the only option.	
-sso	Single sign-on service URL	URL

Option	Description	Values
-ssobinding	SSO Service URL protocol binding	■ REDIR (web SSO) ■ SOAP (ECP)
-type (Required)	Entity type of the export file	saml2idp sam2sp
-username	The CA SiteMinder® Administrator name, which requires the -password option.	string, no default

smfedexport Tool Examples

Example: Exporting an Identity Provider

```
smfedexport -type saml2idp -entityid http://www.myidp.com/idp1
-expiredays 30 -sign -pubkey -slohttpredir http://www.mysite.com
/affwebservices/public/saml2slo -reqsignauthr
-ssoart http://www.mysite.com/affwebservices/public/saml2sso
-artressvc http://www.mysite.com/affwebservices/
saml2artifactresolution -output myidpdescription.xml
```

Example: Exporting a Service Provider

```
smfedexport -type saml2sp -entityid http://www.myidp.com/sp1
-expiredays 30 -sign -pubkey -slohttpredir http://www.mysite.com/
affwebservices/public/saml2slo -signauthr -aconsvcpost
http://www.mysite.com/affwebservices/public/saml2assertionconsumer
-aconsvcpostindex 12345 -output myidpdescription.xml
```

Example: Modifying and Signing an Exported Data File

In this example, you are modifying and digitally signing an XML file using the smfedexport.

To modify and sign a metadata file

1. Edit the existing XML file using an XML editor.
2. Enter the following command:

   ```
   smfedexport -sign -input file -output file
   ```

   For example:

   ```
   smfedexport -sign -input myspdescription.xml -output newspdescription.xml
   ```

To modify an exported file that is already digitally signed

1. Edit the existing XML file using an XML editor as need.
2. Delete the <Signature> element from the file.
3. Enter the following command:

   `smfedexport -sign -input file -output file`

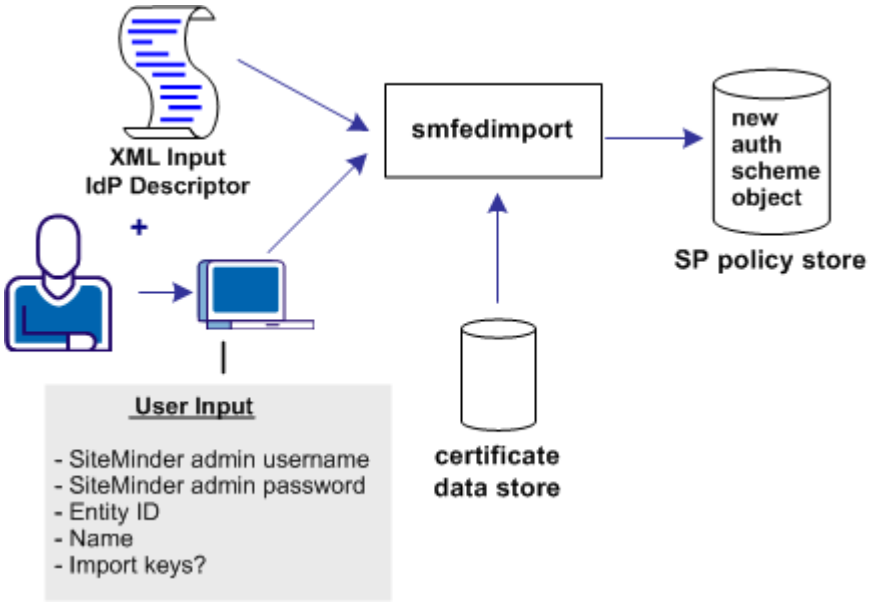
   For example:

   `smfedexport -sign -input myspdescription.xml -output newspdescription.xml`

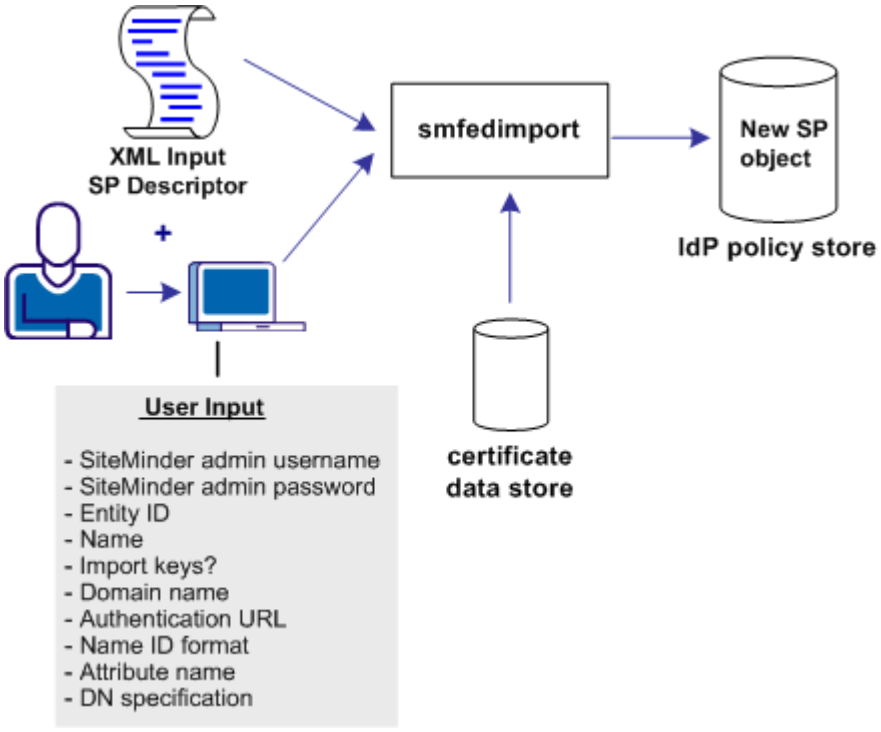
Import Metadata Tool

You can use the import tool for the following tasks:

- Create a SAML 2.0 authentication scheme for a Service Provider, as shown in the following illustration.



- Create a SAML 2.0 Service Provider object for an Identity Provider.



Run the smfedimport Tool

The smfedimport utility can import Identity Providers and Service Providers into a CA SiteMinder® policy store and the certificate data store. If you import a Service Provider input file, the result is a new CA SiteMinder® Service Provider object within an existing affiliate domain. If you import an Identity Provider input file, the result is an authentication scheme that is based on the CA SiteMinder® SAML 2.0 Template.

When the smfedimport command line utility is run, the first and second parameters are the username and password of the CA SiteMinder® administrator. The third and final argument is the path to the input XML file.

To run the smfedimport tool

1. At the system where you installed the Policy Server, open up a command window.
2. Enter the command using the following syntax:

   To import a SAML2 Identity Provider metadata file into the policy store:

   ```
   smfedimport -type saml2idp -username <username>
   -password <password> -entityid <entityid> -name <name>
   [-importkeys <name>] [-silent] -input <file>
   ```

To import a Service Provider metadata file into the policy store:

```
smfedimport -type saml2sp -username <username>
-password <password> -entityid <entityid> -domainname <name>
-authurl <URL> -nameidformat (U|E|X|W|K|N|P|T|U)
-nameidtype  (S | U | D) -attrname <name> -dnspec <spec>
-name <name>[-importkeys <name>] [-silent] -input <file>
```

Note: Switches in square brackets [] are optional.

After smfedimport processes the initial command options, the tool prompts you for additional data based on the type of file you are importing. Any optional arguments that you do not enter on the command line have default values.

smfedimport Tool Examples

Example: Importing Identity Provider metadata

```
smfedimport -type saml2idp -username Siteminder

-password siteminderpassword -entityid http://www.myidp.com
-name mynewauthscheme -importkeys keyaliasname -input mypartnersidpinfo.xml
```

Example: Importing Service Provider metadata

```
smfedimport -type saml2sp -username Siteminder  -password siteminderpassword
-entityid http://www.mysp.com -name mynewsaml2sp -importkeys
keyalisname -domainname myaffiliatedomain
-authurl http://www.mysite.com/login.html -nameidformat U
-nameidtype S -attrname attrname  -input mypartnersspinfo.xml
```

Command Options for smfedimport

The command line options are listed in the following table.

Option	Description	Value
-attrname	Attribute name for the nameID	string
-authurl	Authentication URL	URL
-dnspec	DN specification for the name ID type only	string
-domainname	Affiliate domain name	string

Option	Description	Value
-entityid	Entity ID	The Service Provider ID for the import or the Identity Provider ID for the import
-importkeys	Indicates whether the certificates in the metadata are imported into the certificate data store.	string. Enter a name that becomes an alias for the certificate in the certificate data store. If there are multiple certificates, the aliases are added as name, name1, name2.
-input	Input file	string
-name	Indicates the name of the CA SiteMinder® object, such as the Service Provider name or the SAML authentication scheme name.	string
-nameidformat	Name ID format	(U)nspecificed--default (E)mail address (X)509 Subject name (W)indows domain name (K)erberos Principal Name E(n)tity Identifier (P)ersistent Identifier (T)ransient Identifier
-nameidtype	Name ID type	(S)tatic (U)ser attribute (D)N attribute
-password	CA SiteMinder® Administrator password	string, no default
-type (Required)	Entity type of the import file	saml2idp sam2sp
-silent	Determines whether the tool interactively prompts the user. With this option, the tool operates in silent mode. The tool does not interactively prompt the user for missing input. The tool also does not prompt the user to accept the import of each separate entity in the input file. The tool assumes that all entities in the input file must be imported.	true, if present false otherwise
-username	CA SiteMinder® Administrator name	string, no default

Processing Import Files with Multiple SAML 2.0 Providers

If multiple providers are specified in one import file, the tool imports them into the same affiliate domain. The names for each provider are based on the value you specify for the smfedimport command option -**name**.

For example, if there are three Service Providers in the import file and you specify:

```
-name mySP
```

The tool registers the imported providers as mysp, mysp_1, and mysp_2. The integer increases by one for each subsequent provider. If there is a mixture of Identity Providers and Service Providers in an import file, the naming convention still applies.

Processing Import Files with Multiple Certificate Aliases

If there are multiple certificates in the import file, the tool imports them into the certificate data store. The tool then assigns alias names from the value you specify with that smfedimport command option **–importkeys**.

For example, if there are three certificates in the import file and you specify:

```
-importkeys myalias
```

The tool registers the imported certificates as myalias, myalias_1, and myalias_2. The integer increases by one for each subsequent certificate.

Chapter 21: Legacy Federation Trace Logging

This section contains the following topics:

Trace Logging

The Web Agent trace logging facility and the Policy Server Profiler enable CA SiteMinder® to monitor the performance of the Web Agent and Policy Server. These logging mechanisms provide comprehensive information about CA SiteMinder® processes so you can analyze the performance and troubleshoot the issues.

For legacy federation, several logging components are available to collect trace messages about federated communication. Trace messages provide detailed information about program operation for tracing, debugging, or both. Trace messages are ordinarily turned off during normal operation. You can enable them to extract in-depth information in addition to the trace message itself. For example, you can look at the FWSTrace.log to see the SAML assertion that CA SiteMinder® generates or to collect the name of the current user.

The collected trace messages are written to a trace log. The FWSTrace.log is located in the directory *web_agent_home*/log.

Note: For Web Agents on IIS 6.0 servers, log files are created only after the first user request has been submitted. To verify your configuration in the log file, a user has to submit a request.

You can establish trace logs at the Web Agent and the Policy Server to monitor CA SiteMinder® operation.

Flush FWS Cache for Trace Logs

If you modify the federation configuration at the asserting or relying party, flush the Federation Web Services cache for the changes to appear in the trace logs. An example of modifying the configuration is to enable or disable a SAML binding.

Follow these steps:

1. Log in to the Administrative UI.
2. Select Administration, Policy Server, Cache Management.
3. Click Flush All in the All Caches section.
4. Click Close.

All caches are now cleared.

Log Messages for the Fed_Client Component

The Web Agent Option Pack installs the Federation Web Services (FWS) application. The FWS application represents the federation client. The component that controls the trace messages and monitors FWS activity is the Fed_Client component.

FWS uses the common tracing facility that the Web Agent uses to log trace messages. The following files set up trace logging:

trace configuration file

Specifies the configuration file that determines which components and events FWS monitors. The default file is fwstrace.conf.

trace log file

Specifies the output file for all the logged messages. You provide a name and the location for this file in the Web Agent configuration file.

Web Agent Configuration File or Agent Configuration Object

Contains the logging parameters that enable logging and format the log. This file does not define message content.

The Fed_Client component includes the following sub components:

single sign-on

Monitors single sign-on activity.

single logout

Monitors requests for single logout.

discovery profile

Monitors the identity provider discovery profile activity.

administration

Watches administration-related messages.

request

Monitors request and authentication activity.

general

Monitors activity that other subcomponents are not monitoring.

configuration

Monitors SAML 2.0 Service Provider configuration messages.

Configure FWS Trace Logging

To collect trace messages for the Federation Web Services application, configure the FWS trace logging.

Follow these steps:

1. Do one of the following tasks:
   - Make a copy of the default template, FWSTrace.conf and modify the file to include only the data you want to monitor.
   - Copy one of the preconfigured templates and assign a new name to it.

   Note: Do not edit the template directly.
2. Open the LoggerConfig.properties file in the directory *web_agent_home*/affwebservices/WEB-INF/classes, and set the following parameters:
   - Set TracingOn to Yes. This option instructs the trace facility to write messages to a file.
   - Set the TraceFileName parameter to the full path of the trace log file. The default location is in *web_agent_home*/config/FWSTrace.log.
   - Set the TraceConfigFile parameter to the full path of the trace configuration file, either the default template, FWSTrace.conf or another template. Templates can be found at *web_agent_home*/config.

3. Optionally, you can format the trace log file, the file that contains the log output. The following parameters are the Web Agent configuration parameters that dictate the format of the trace log file:

   - TraceRollover
   - TraceSize
   - TraceCount
   - TraceFormat
   - TraceDelim

   The LoggerConfig.properties file contains descriptions of all these settings.

Log Messages for the Fed_Server Component

The component that controls the trace messages for federation services at the Policy Server is the Fed_Server component. This component monitors activity for the assertion generator and the SAML authentication scheme. For example, you can view the generated assertion in the smtracedefault.log file.

To configure logging at the Policy Server, use the Policy Server Profiler. The Profiler lets you specify components for trace logging, which include.

For more information about using the profiler, see the *CA SiteMinder® Policy Server Administration Guide*.

Federation Services Trace Logging (smtracedefault.log)

The profiler is the Policy Server facility for logging. You can use the profiler to collect trace messages for federation services and write them to the smtracedefault.log file.

The component that controls the trace messages for federation services at the Policy Server is the Fed_Server component.

The Policy Server Profiler allows you to trace internal Policy Server diagnostics and processing functions.

Follow these steps:

1. Start the Policy Server Management Console.

   Important! If you are accessing this graphical user interface on Windows Server 2008, open the shortcut with Administrator permissions. Use Administrator permissions even if you are logged in to the system as an Administrator. For more information, see the release notes for your CA SiteMinder® component.

2. Click the Profiler tab.

3. Set the Enable Profiling option to enable profiling.
4. To select configuration settings for the Profiler, do one of the following:
   - Accept the Profiler settings specified by the default smtracedefault.txt file presented in the Configuration File drop-down list.
   - Select another configuration file that has already been selected during this management session from the Configuration File drop-down list.
   - Click the Browse button to select another configuration file.
5. To change the Profiler settings stored in a Profiler configuration file and save them in the same or a new file, click the Configure Settings button to open the Policy Server Profiler dialog.
6. Adjust the settings presented in the Output group box to specify the output format for information generated by the Policy Server Profiler.
7. Click Apply to save your changes.

Notes:

Changes to the Profiler settings take effect automatically. However, if you restart the Policy Server, a new output file (if the Profiler is configured for file output) is created. The existing Profiler output file is automatically saved with a version number. For example:

```
smtracedefault.log.1
```

If changes to the Logging or Tracing facility settings are not related to the Profiler output file, for example, enabling/disabling the console logging on Windows, the existing file is appended with new output without saving a version of the file.

By default The Policy Server retains up to ten output files (the current file and nine backup files). Older files are replaced automatically with newer files when the ten file limit is reached. You can change the number of files to retain by configuring the TraceFilesToKeep DWORD registry setting to the required decimal value. The TraceFilesToKeep registry setting must be created in the following registry location:

```
HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\Netegrity\SiteMinder\CurrentVersion\
LogConfig\TraceFilesToKeep
```

The Profiler tab has a "Buffered Tracing" option, which is set by default to improve Policy Server performance. This option is on Solaris systems only.

Update FWS Data in the Logs

If you modify any part of the federation configuration, flush the Federation Web Services cache for the changes to appear in the trace logs.

Note: A brief delay can occur from when the changes are made and when Federation Web Services receives the information.

Follow these steps:

1. Log in to the Administrative UI.
2. Click Administration, Policy Server, Cache Management.
3. Click Flush All in the All Caches section of the page.
4. Click Close.

Simplify Logging with Trace Configuration Templates

To make the task of collecting tracing data simpler, a series of preconfigured templates are installed with the Policy Server and the Web Agent Option Pack. You can use these templates instead of creating your own trace configuration file to collect the data that gets written to a trace log.

Trace Logging Templates for FWS

The following templates are available for Federation Web Services:

Template	Tracing Messages Collected
WebAgentTrace.conf	Default template. Collects data that you specify.
FWS_SSOTrace.conf	Collects single sign-on messages
FWS_SLOTrace.conf	Collects single logout messages
FWS_IPDTrace.conf	Collects Identity Provider Discovery Profile messages

All these templates include the Fed_Client component and subcomponents for the specific data being tracked. Look at each template to see the exact contents. The templates are located in *web_agent_home*/config.

To use a template for trace logging

1. Make a copy of the template you want to use and rename the copy.

   Note: Do not edit the template directly.

2. Open the Agent configuration file or Agent configuration Object.
3. Set the TraceFile parameter to Yes.
4. Set the TraceFileName parameter to the full path to the trace log file. This file contains the log output.
5. Set the TraceConfigFile parameter to the full path to the newly named template file.
6. Format the trace log file. The following parameters are the Web Agent configuration parameters that dictate the format of the trace log file:

   - TraceAppend
   - TraceFormat
   - TraceDelimiter
   - TraceFileSize
   - LogLocalTime

   For descriptions of each logging parameter, see the *Web Agent Configuration Guide*.

Note: Web Agents on IIS 6.0 and Apache 2.0 servers do not support dynamic configuration of log parameters that are set locally in the Agent configuration file. Consequently, when you modify a parameter, the change takes effect only after the Agent is restarted. If you configure the log parameters in an Agent configuration object, these log settings can be stored and updated dynamically.

FWS Template Sample

The following text is an excerpt from the FWS_SLOTrace.conf template. Most of the file contains comments and instructions on how to use the file, the command syntax, and the available subcomponents for the Fed_Client component.

The excerpt shows the component, Fed_Client and the subcomponents (Single_Logout and Configuration) that are monitored. The excerpt also shows the specific data fields that indicate the required contents of each message (Date, Time, Pid, Tid, TransactionId, SrcFile, Function, Message).

```
components: Fed_Client/Single_Logout, Fed_Client/Configuration
data: Date, Time, Pid, Tid, TransactionID, SrcFile, Function, Message
```

Trace Logging Templates for the IdP and SP

To make the task of collecting tracing data simpler, a series of preconfigured templates are installed with the Policy Server. You can use these templates instead of creating your own trace configuration file to collect the data that gets written to a trace log.

The following templates are available for trace logging related to the Identity Provider and the Service Provider, such as assertion generation or SAML authentication.

Template	Tracing Messages Collected
samlidp_trace.template	Collects messages for Identity Provider activity
samlsp_trace.template	Collects messages for Service Provider activity

Look at each template to see the exact contents. The templates are located in *siteminder_home*/config/profiler_templates.

Service Provider Template Sample

The following text is the samlsp_trace.template file.

```
components: Server/Policy_Server_General, IsProtected/Resource_Protection,
Login_Logout/Authentication, Login_Logout/Policy_Evaluation,
Login_Logout/Active_Expression, Login_Logout/Session_Management,
IsAuthorized/Policy_Evaluation, JavaAPI, Fed_Server/Auth_Scheme,
Fed_Server/Configuration
data: Date, Time, Tid, TransactionID, SrcFile, Function, Domain, Resource, Action,
User, Message
```

For legacy federation, it includes the Fed_Server component along with the subcomponents Auth_Scheme and Configuration.

The data fields that indicate the required contents of each message are:

Date, Time, Tid, TransactionId, SrcFile, Function, Domain, Resource, Action User, and Message.

Identity Provider Profiler Sample

At the Identity Provider, the Profiler tab of the Policy Server Management Console specifies a template in the Configuration File field. The following text is a sample entry for the Configuration File field:

c:\program files\ca\siteminder\config\profile_templates\samlidp_template.trace

For more information about using the Profiler, see the *Policy Server Administration Guide*.

Chapter 22: Configuration Settings that Must Use the Same Values

This section contains the following topics:

How to Use the Configuration Settings Tables

When configuring a federated environment, there are many instances where you must configure matching parameter values at both sides of a transaction.

The tables that follow explicitly describe each matching set of parameters. Each cell in a row describes a setting that must match the corresponding value or values described in the other cells in the row.

Note: The information is only applicable in an environment where the asserting and relying party are CA SiteMinder® systems.

SAML 1.x Matching Configuration Settings

The following table lists CA SiteMinder® configuration settings that you must set to the same value at the SAML 1.x producer and consumer. The table also indicates the dialog or file where these settings are located. Most of these settings are in the Administrative UI; however, some parameters are in a properties file or part of a link.

- The first column describes a setting that you must configure in the Administrative UI at the Consumer.
- The second column describes a setting that you must configure in the Administrative UI at the Producer and must match the setting at the Consumer.

Important! If you have to enter a URL, the URL string that comes after the colon is case-sensitive. For example, all text that follows **http:** is case-sensitive. Therefore, the case of the URLs in all Audience-related settings and Assertion Consumer URL-related settings must match.

These Settings at the SAML 1.x Consumer...	Must Match These Settings at the SAML 1.x Producer...
Affiliate Name Scheme Setup section of the authentication scheme page (Artifact and POST profiles)	**Name** field General settings for the affiliate object Value must be lowercase **NAME** query parameter in intersite transfer URL links at the producer.
Password field (SAML Artifact auth. scheme only) Scheme Setup section of the authentication scheme page	**Password/Confirm Password** fields General settings for the affiliate object
Audience field Any other SAML consumer; Scheme Setup section of the authentication scheme page	**Audience** field Assertions settings for the affiliate object
Assertion Consumer URL (SAML POST auth. scheme only) Scheme Setup section of the authentication scheme page	**Assertion Consumer URL** Assertions settings for the affiliate object **SMCONSUMERURL** query parameter intersite transfer URL links at the producer
Issuer field Scheme Setup section of the authentication scheme page	**AssertionIssuerID** parameter AMAssertionGenerator.properties file at the producer
Version from the SAML Version drop-down list Scheme Setup section-- authentication scheme page (SAML Artifact auth. scheme only)	**Version from the SAML Version drop-down list** Assertions settings for the affiliate object.
Company Source ID Scheme Setup section-- authentication scheme page (SAML Artifact auth. scheme only)	**SourceID** parameter AMAssertionGenerator.properties file at the producer

SAML 2.0 Matching Configuration Settings

The following table lists CA SiteMinder® configuration settings that you must set to the same value at the SAML 2.0 Identity Provider and Service Provider. The table also indicates where these settings are located. Most of these settings are in the Administrative UI; however, some parameters are in a properties file or part of a link.

- The first column describes a setting that you must configure in the Administrative UI at the Service Provider.
- The second column describes a setting that you must configure in the Administrative UI at the Identity Provider and must match the setting at the Service Provider.

Important! If you have to enter a URL, the URL string that comes after the colon is case-sensitive. For example, text following **http:** is case-sensitive. Therefore, the case of all SP ID- and IdP ID-related settings must match.

These Settings at the Service Provider...	Must Match These Settings at the Identity Provider...
Attribute Name Add/Edit Attribute page from the Attributes settings of the SAML 2.0 authentication scheme.	**Variable Name** Attribute Setup section of the Add Attribute page from the Attributes settings for the SAML Service Provider object.
Audience field ■ Any other SAML Service Provider ■ SSO settings of the SAML 2.0 authentication scheme.	**Audience** field SSO section of the SAML Profiles settings for the SAML Service Provider object.
IdP ID field General settings of the SAML 2.0 authentication scheme	**IdP ID** field ■ General settings for the SAML Service Provider object ■ For Identity Provider-initiated SSO--**SPID** query parameter in an unsolicited response
Local Name Add/Edit Attribute page from the Attributes settings of the SAML 2.0 authentication scheme. **Local Name** Federation Attribute Variable page for creating a Federation Attribute variable at the SAML Requester (Service Provider).	None

These Settings at the Service Provider...	Must Match These Settings at the Identity Provider...
SP ID field ■ General settings of the SAML 2.0 authentication scheme ■ For Service Provider-initiated SSO-- **ProviderID** query parameter in hard-coded links to the Identity Provider	**SP ID** field General settings for the SAML Service Provider object
SP Name Backchannel section of the Encryption & Signing settings of the SAML 2.0 authentication scheme. This value must be in lowercase.	**Name** field General settings for the SAML Service Provider object This value must be in lowercase.

WS-Federation Configuration Settings

The following table lists CA SiteMinder® configuration settings that you must set to the same value at the WS-Federation Account Partner and Resource Partner. Read the table as follows:

- The first column describes a setting that you must configure in the Administrative UI at the Resource Partner.
- The second column describes a setting that you must configure in the Administrative UI at the Account Partner and must match the setting at the consumer.

Important! If you have to enter a URL, the URL string that comes after the colon is case-sensitive. For example, any text that follows **http:** is case-sensitive. Therefore, the case of all RP ID- and AP ID-related settings must match.

These Settings at the Resource Partner...	Must Match These Settings at the Account Partner...
Resource Partner ID General settings for the WS-Federation authentication scheme	**Resource Partner ID** General settings of the Resource Partner object wtrealm query parameter must be set to the Resource Partner ID for the hard-coded link to trigger Account Partner-initiated SSO.

These Settings at the Resource Partner...	Must Match These Settings at the Account Partner...
Account Partner ID General settings for the WS-Federation authentication scheme	**Account Partner ID** General settings of the Resource Partner object

Chapter 23: Federation Web Services URLs Used by SiteMinder

This section contains the following topics:

Federation Services URLs

The Federation Web Services contains many services to implement legacy federation. When configuring single sign-on, single logout, or identity provider discovery profile through the Administrative UI, you are required to specify URLs that reference the different services.

The following service descriptions include:

- A brief description of the service
- The URL for the service
- The field in the Administrative UI where you enter the URL
- Associated servlet and servlet mapping in the Web.xml file

The Web.xml file is one of the deployment descriptors for the Federation Web Services application. This file lists servlets and URL mappings.

URLs for Services at the Asserting Party

The following services are provided at the asserting party (Producer/Identity Provider/Account Partner); however, you enter the service URL at the relying party (Consumer/Service Provider/Resource Partner).

The Federation Web Services application supplies the following services:

- Intersite Transfer Service (see page 368) (SAML 1.x producer)
- Assertion Retrieval Service (see page 369) (SAML 1.x producer)
- Artifact Resolution Service (see page 370) (SAML 2.0 IdP)
- Single sign-on Service (see page 371) (SAML 2.0 IdP)

- Single logout Service (see page 373) (SAML 2.0 IdP)
- Identity Provider Discovery Profile Service (see page 375) (SAML 2.0)
- Attribute Service (see page 376) (SAML 2.0)
- Single sign-on Service (see page 372) (WS-Federation)
- Signout Service (see page 374) (WS-Federation)
- WSFedDispatcher Service (see page 377) (WS-Federation)

Intersite Transfer Service URL (SAML 1.x)

For SAML 1.x POST and artifact profiles, the intersite transfer URL is a producer-side component that transfers a user from the producer to the consumer.

Default URL for this Service

http://*producer_server:port*/affwebservices/public/intersitetransfer

producer_server:port

Identifies the web server and port number of the system at the producer hosting the Web Agent Option Pack or the SPS federation gateway.

Intersite Transfer URL

Include the URL in a hard-coded link on a page at the producer.

Associated Servlet and Servlet Mapping in the Web.xml file

```
<servlet>
  <servlet-name>intersiteTransferService</servlet-name>
  <display-name>Intersite Transfer Service</display-name>
  <description>This servlet acts as the Intersite Transfer URL.</description>
  <servlet-class>com.netegrity.affiliateminder.webservices.
    IntersiteTransferService
  </servlet-class>
</servlet>


<servlet-mapping>
  <servlet-name>intersiteTransferService</servlet-name>
  <url-pattern>/public/intersitetransfer/*</url-pattern>
</servlet-mapping>
```

Assertion Retrieval Service URL (SAML 1.x)

The Assertion Retrieval Service retrieves an assertion for a SAML. 1.x consumer site.

Default URLs for this Service

- For Basic or Basic over SSL to protect this service, the URL is:
  https://producer_server:port/affwebservices/assertionretriever
- For client certificate authentication to protect this service, the URL is:
  https://producer_server:port/affwebservices/certassertionretriever

producer_server:port

Identifies the web server and port number of the system at the producer hosting the Web Agent Option Pack or the SPS federation gateway.

Assertion Retrieval URL

Specified in the Assertion Retrieval URL field. This field is in the Scheme Setup section of the SAML 1.x authentication scheme page.

Associated Servlet and Servlet Mapping in the Web.xml file

```
<servlet>
  <servlet-name>assertionretriever</servlet-name>
  <display-name>SAML Assertion Retrieval servlet</display-name>
  <description>This servlet processes the HTTP post based SAML requests and
returns the SAML Response elements. Both SAML Request and Response elements are
SOAP encoded.</description>
  <servlet-class>com.netegrity.affiliateminder.webservices.
     AssertionRetriever</servlet-class>
</servlet>


<servlet-mapping>
  <servlet-name>assertionretriever</servlet-name>
  <url-pattern>/assertionretriever/*</url-pattern>
</servlet-mapping>
<servlet-mapping>

  <servlet-name>assertionretriever</servlet-name>
  <url-pattern>/certassertionretriever/*</url-pattern>
</servlet-mapping>
```

Artifact Resolution Service URL (SAML 2.0)

The Artifact Resolution Service retrieves SAML 2.0 assertions for a Service Provider.

Default URL for this Service

- For Basic authentication to protect this service, the URL is:

  http://*idp_server:port*/affwebservices/saml2artifactresolution

- For Basic over SSL or X.509 client certificate authentication to protect this service, the URL is:

  https://*idp_server:port*/affwebservices/saml2certartifactresolution

idp_server:port

Identifies the web server and port hosting the Web Agent Option Pack or SPS federation gateway.

Resolution Service URL

Specified in the Resolution Service field. This field is in the Bindings section of the SSO settings for the SAML 2.0 authentication scheme. To make the field active, select HTTP-Artifact as the binding.

Associated Servlet and Servlet Mapping in the Web.xml file

```
<servlet>
  <servlet-name>saml2artifactresolution</servlet-name>
  <display-name>SAML 2.0 Single Sign-On service</display-name>
  <description>This servlet is the SAML 2.0 Artifact Resolution
      service at an IdP.</description>
  <servlet-class>com.netegrity.affiliateminder.webservices.
    saml2.ArtifactResolution</servlet-class>
</servlet>


<servlet-mapping>
<servlet-name>saml2artifactresolution</servlet-name>
<url-pattern>/saml2artifactresolution/*</url-pattern>
</servlet-mapping>


<servlet-mapping>
<servlet-name>saml2artifactresolution</servlet-name>
<url-pattern>/saml2certartifactresolution/*</url-pattern>
</servlet-mapping>
```

Single Sign On Service URL (SAML 2.0)

The single sign-on service implements single sign-on for SAML 2.0.

Default URL for this Service

http://*idp_server:port*/affwebservices/public/saml2sso

idp_server:port

Identifies the web server and port hosting the Web Agent Option Pack or SPS federation gateway.

SSO Service URL

Specified in the SSO Service field. This field is in the SSO settings for the SAML 2.0 authentication scheme.

Associated Servlet and Servlet Mapping in the Web.xml file

```
<servlet>
  <servlet-name>saml2sso</servlet-name>
  <display-name>SAML 2.0 Single Sign-On service</display-name>
  <description>This servlet is the SAML 2.0 Single Sign-On service at an
IdP.</description>
  <servlet-class>com.netegrity.affiliateminder.webservices.
saml2.SSO</servlet-class>
</servlet>


<servlet-mapping>
  <servlet-name>saml2sso</servlet-name>
  <url-pattern>/public/saml2sso/*</url-pattern>
</servlet-mapping>
```

Single Sign-on Service URL (WS-Federation)

The WS-Federation single sign-on service implements single sign-on for WS-Federation.

Default URL for this Service

http://*ap_server:port*/affwebservices/public/wsfedsso

ap_server:port

Specifies the server and port number of the system at the Account Partner. The system is hosting the Web Agent Option Pack or the SPS federation gateway, depending on which component is installed in your federation network.

SSO Service URL

Specified in the SSO Service field. This field is in the SSO settings of the WS-Federation authentication scheme.

Associated Servlet and Servlet Mapping in the Web.xml file

```
<servlet>
<servlet-name>wsfedsso</servlet-name>
<display-name>WSFED Single Sign-On service</display-name>
<description>This servlet is the WSFED Single Sign-On service at an Account
Partner.</description>
<servlet-class>com.netegrity.affiliateminder.webservices.wsfed.SSO
  </servlet-class>
</servlet>

<servlet-mapping>
<servlet-name>wsfedsso</servlet-name>
<url-pattern>/public/wsfedsso/*</url-pattern>
</servlet-mapping>
```

Single Logout Service URL at the IdP (SAML 2.0)

This service implements single logout for SAML 2.0.

Default URL for this Service

http://*idp_server:port*/affwebservices/public/saml2slo

idp_server:port

Identifies the web server and port hosting the Web Agent Option Pack or SPS federation gateway.

SLO Location URL/SLO Response Location URL

Specified in the fields of the same name at the Identity Provider. These fields are in the SLO section of the SAML Profiles settings for the SAML Service Provider object.

Associated Servlet and Servlet Mapping in the Web.xml file

```
<servlet>
  <servlet-name>saml2slo</servlet-name>
  <display-name>SAML 2.0 Single Logout service</display-name>
  <description>This servlet is the SAML 2.0 Single Logout service at an
IdP.</description>
  <servlet-class>com.netegrity.affiliateminder.webservices.
     saml2.SLOService</servlet-class>
</servlet>

<servlet-mapping>
  <servlet-name>saml2slo</servlet-name>
  <url-pattern>/public/saml2slo/*</url-pattern>
</servlet-mapping>
```

Signout Service URL at the AP (WS-Federation)

This signout service implements WS-Federation sign out functionality.

Default URL for this Service

http://*ap_server:port*/affwebservices/public/wsfedsignout

ap_server:port

Specifies the server and port number of the system at the Account Partner. The system is hosting the Web Agent Option Pack or the SPS federation gateway, depending on which component is installed in your federation network.

Signout Cleanup URL/Signout Confirm URL

Specified in fields of the same name at the Account Partner. These fields are in the Signout section of the SAML Profiles settings for the Resource Partner Properties object.

Associated Servlet and Servlet Mapping in the Web.xml file

```
<servlet>
  <servlet-name>wsfedsignout</servlet-name>
  <display-name>WS-Federation Signout Service</display-name>
  <description>This servlet is the WS-Federation Signout service
      at an AP.</description>
  <servlet-class>com.netegrity.affiliateminder.webservices.wsfed.
    SignoutService</servlet-class>
</servlet>


<servlet-mapping>
<servlet-name>wsfedsignout</servlet-name>
<url-pattern>/public/wsfedsignout/*</url-pattern>
</servlet-mapping>
```

Identity Provider Discovery Profile Service URL (SAML 2.0)

The Identity Provider Discovery Profile service implements the Identity Provider Discovery feature.

Default URL for this Service

https://*idp_server:port*/affwebservices/public/saml2ipd/*

idp_server:port

Identifies the web server and port hosting the Web Agent Option Pack or SPS federation gateway.

Service URL

Specified in the Service URL field. This field is located in the IPD section of the SAML Profile settings for the SAML Service Provider object at the Identity Provider.

Associated Servlet and Servlet Mapping in Web.xml file

```
<servlet>
  <servlet-name>saml2ipd</servlet-name>
  <display-name>SAML 2.X Identity Provider Discovery Profile
    service</display-name>
  <description>This servlet is the SAML 2.X Identity Provider Discovery Profile
service at an SP or IdP.</description>
  <servlet-class>com.netegrity.affiliateminder.webservices.
    saml2.IPDService</servlet-class>
</servlet>


<servlet-mapping>
  <servlet-name>saml2ipd</servlet-name>
  <url-pattern>/public/saml2ipd/*</url-pattern>
</servlet-mapping>
```

Attribute Service URL (SAML 2.0)

The Attribute Service enables an Attribute Authority to respond to attribute queries from a SAML Requester.

Default URL for this Service

http://*idp_server:port*/affwebservices/saml2attributeservice

idp_server:port

Identifies the web server and port hosting the Web Agent Option Pack or SPS federation gateway.

Attribute Service URL

Specified in the Attribute Service field. This field is in the Attributes settings for the SAML 2.0 authentication scheme at the Service Provider.

Associated Servlet and Servlet Mapping in the Web.xml file

```
<servlet>
  <servlet-name>saml2attributeservice</servlet-name>
  <display-name>SAML 2.0 Attribute service</display-name>
  <description>This servlet is the SAML 2.0 Attribute Service
        at an IdP.</description>
  <servlet-class>com.netegrity.affiliateminder.webservices.saml2.
      AttributeService</servlet-class>
</servlet>


<servlet-mapping>
 <servlet-name>saml2attributeservice</servlet-name>
 <url-pattern>/saml2attributeservice/*</url-pattern>
</servlet-mapping>

<servlet-mapping>
 <servlet-name>saml2attributeservice</servlet-name>
 <url-pattern>/saml2certattributeservice/*</url-pattern>
</servlet-mapping>
```

WSFedDispatcher Service URL at the AP

The WSFedDispatcher Service receives all incoming WS-Federation messages and forwards the request processing to other services based on the query parameter data.

Default URL for this Service

https://*ap_server:port*/affwebservices/public/wsfeddispatcher

ap_server:port

Specifies the server and port number of the system at the Account Partner. The system is hosting the Web Agent Option Pack or the SPS federation gateway, depending on which component is installed in your federation network.

URL

Not applicable

Associated Servlet and Servlet Mapping in the Web.xml file

```
<servlet>
  <servlet-name>wsfeddispatcher</servlet-name>
  <display-name>WS-Federation Dispatcher service</display-name>
  <description>This servlet is the WS-Federation Dispatcher service for all
WS-Federation services.</description>
  <servlet-class>com.netegrity.affiliateminder.webservices.wsfed.
     dispatcher</servlet-class>
</servlet>


<<servlet-mapping>
<servlet-name>wsfeddispatcher</servlet-name>
<url-pattern>/public/wsfeddispatcher/*</url-pattern>
</servlet-mapping>
```

URLs for Services at the Relying Party

The relying party provides the following services; however, you enter the URL for the service at the asserting party.

The CA SiteMinder® relying party provides the following services:

- SAML credential collector (SAML 1.x) (see page 378)
- AuthnRequest service (SAML 2.0) (see page 379)
- Assertion Consumer Service (SAML 2.0) (see page 380)
- Security Token Consumer Service (WS-Federation) (see page 381)

- Single Logout Service (SAML 2.0) (see page 382)
- Signout Service (WS-Federation) (see page 383)
- WSFedDispatcher Service (WS-Federation) (see page 384)

SAML Credential Collector Service URL (SAML 1.x)

The SAML Credential Collector service assists in consuming SAML 1.x assertions.

Default URL for this Service

https://*consumer_server:port*/affwebservices/public/samlcc

consumer_server:port

Identifies the web server and port hosting the Web Agent Option Pack or SPS federation gateway.

Assertion Consumer URL

Specified in the Assertion Consumer URL field. This field is on the Assertions page for the SAML 1.x affiliate object. The field is also in the Scheme Setup section for the SAML 1.x POST authentication scheme at the consumer.

Associated Servlet and Servlet Mapping in the Web.xml file

```
<servlet>
  <servlet-name>samlcredentialcollector</servlet-name>
  <display-name>SAML Credential Collector</display-name>
  <description>This servlet acts as the SAML Credential Collector.</description>
  <servlet-class>com.netegrity.affiliateminder.webservices.
     SAMLCredentialCollector</servlet-class>
</servlet>

<servlet-mapping>
  <servlet-name>samlcredentialcollector</servlet-name>
  <url-pattern>/public/samlcc/*</url-pattern>
</servlet-mapping>
```

AuthnRequest Service (SAML 2.0)

This AuthnRequest service helps implement single sign-on for the artifact or POST profile.

Default URL for this Service

https://*sp_server:por*t/affwebservices/public/saml2authnrequest

sp_server:port

Specifies the server and port number at the Service Provider hosting the Web Agent Option Pack or the SPS federation gateway.

URL for the Service

Not applicable.

The AuthnRequest is a link in an application at the Service Provider. This link initiates single sign-on and it must be in an application.

Associated Servlet and Servlet Mapping in the Web.xml file

```
<servlet>
  <servlet-name>saml2authnrequest</servlet-name>
  <display-name>SAML 2.0 AuthnRequest service</display-name>
  <description>This servlet is the SAML 2.0 AuthnRequest service at an
SP.</description>
  <servlet-class>com.netegrity.affiliateminder.webservices.
   saml2.AuthnRequest</servlet-class>
</servlet>

<servlet-mapping>
  <servlet-name>saml2authnrequest</servlet-name>
  <url-pattern>/public/saml2authnrequest/*</url-pattern>
</servlet-mapping>
```

Assertion Consumer Service URL (SAML 2.0)

The Assertion Consumer Service enables the consumption of assertions.

Default URL for this Service

https://*sp_server:port*/affwebservices/public/saml2assertionconsumer

sp_server:port

Specifies the server and port number at the Service Provider hosting the Web Agent Option Pack or the SPS federation gateway.

Assertion Consumer URL

Specified in the Assertion Consumer URL field. This field is part of the SSO settings for the SAML Service Provider object at the Identity Provider.

Associated Servlet and Servlet Mapping in the Web.xml file

```
<servlet>
  <servlet-name>saml2assertionconsumer</servlet-name>
  <display-name>SAML 2.0 Assertion Consumer service</display-name>
  <description>This servlet is the SAML 2.0 Assertion Consumer service at an
SP.</description>
  <servlet-class>com.netegrity.affiliateminder.webservices.
    saml2.AssertionConsumer</servlet-class>
</servlet>


<servlet-mapping>
  <servlet-name>saml2assertionconsumer</servlet-name>
  <url-pattern>/public/saml2assertionconsumer/*</url-pattern>
</servlet-mapping>
```

Security Token Consumer Service URL (WS-Federation)

The Security Token Consumer Service enables the consumption of assertions at the Resource Partner.

Default URL for this Service

https://*rp_server:port*/affwebservices/public/wsfedsecuritytokenconsumer

rp_server:port

Identifies the web server and port at the Resource Partner hosting the Web Agent Option Pack or SPS federation gateway.

Security Token Consumer Service URL

Specified in the Security Token Consumer Service field. This field is part of the SAML Profiles settings for the Resource Partner object at the Account Partner.

Associated Servlet and Servlet Mapping in the Web.xml file

```
<servlet>
  <servlet-name>wsfedsecuritytokenconsumer</servlet-name>
  <display-name>Security Token Consumer service</display-name>
  <description>This servlet is the WS-Federation Security Token
     Consumer service at an RP.</description>
  <servlet-class>com.netegrity.affiliateminder.webservices.wsfed.
     SecurityTokenConsumer</servlet-class>
</servlet>


<<servlet-mapping>
<servlet-name>wsfedsecuritytokenconsumer</servlet-name>
<url-pattern>/public/wsfedsecuritytokenconsumer/*</url-pattern>
</servlet-mapping>
```

Single Logout Service URL at the SP (SAML 2.0)

The single logout services implement single logout for SAML 2.0.

Default URL for this Service

http://*sp_server:port*/affwebservices/public/saml2slo

sp_server:port

Specifies the server and port number at the Service Provider hosting the Web Agent Option Pack or the SPS federation gateway.

SLO Location URL/SLO Response Location URL

Specified in the fields of the same name. These fields are part of the SLO settings for the SAML 2.0 authentication scheme that you configure at the Service Provider.

Associated Servlet and Servlet Mapping in the Web.xml file

```
<servlet>
  <servlet-name>saml2slo</servlet-name>
  <display-name>SAML 2.0 Single Logout service</display-name>
  <description>This servlet is the SAML 2.0 Single Logout service at an
SP.</description>
  <servlet-class>com.netegrity.affiliateminder.webservices.
     saml2.SLOService</servlet-class>
</servlet>

<servlet-mapping>
  <servlet-name>saml2slo</servlet-name>
  <url-pattern>/public/saml2slo/*</url-pattern>
</servlet-mapping>
```

Signout Service URL at the RP (WS-Federation)

The Signout service implements sign out functionality for WS-Federation.

Default URL for this Service:

http://*rp_server:port*/affwebservices/public/wsfedsignout

rp_server:port

Identifies the web server and port at the Resource Partner hosting the Web Agent Option Pack or SPS federation gateway.

Signout Cleanup URL/Signout URL

Specified in fields of the same name. These fields are in the Signout section for the WS-Federation authentication scheme at the Resource Partner.

Associated Servlet and Servlet Mapping in the Web.xml file

```
<servlet>
  <servlet-name>wsfedsignout</servlet-name>
  <display-name>WS-Federation Signout Service</display-name>
  <description>This servlet is the WS-Federation Signout service
      at an RP.</description>
  <servlet-class>com.netegrity.affiliateminder.webservices.wsfed.
    SignoutService</servlet-class>
</servlet>


<servlet-mapping>
<servlet-name>wsfedsignout</servlet-name>
<url-pattern>/public/wsfedsignout/*</url-pattern>
</servlet-mapping>
```

WSFedDispatcher Service URL at the RP

The WSFedDispatcher Service receives all incoming WS-Federation messages. The service then forwards the request processing to other services based on the query parameter data.

Default URL for this Service

https://*rp_server:port*/affwebservices/public/wsfeddispatcher

rp_server:port

Identifies the web server and port at the Resource Partner hosting the Web Agent Option Pack or SPS federation gateway.

URL for Service

Not applicable.

Associated Servlet and Servlet Mapping in the Web.xml file

```
<servlet>
  <servlet-name>wsfeddispatcher</servlet-name>
  <display-name>WS-Federation Dispatcher service</display-name>
  <description>This servlet is the WS-Federation Dispatcher service for all
WS-Federation services.</description>
  <servlet-class>com.netegrity.affiliateminder.webservices.wsfed.
     dispatcher</servlet-class>
</servlet>


<<servlet-mapping>
<servlet-name>wsfeddispatcher</servlet-name>
<url-pattern>/public/wsfeddispatcher/*</url-pattern>
</servlet-mapping>
```

The Web.xml File

The Web.xml file lists servlets and URL mappings for the Federation Web Services application.

You cannot change most of this file, but you can modify the URL mappings.

To view the Web.xml file, go to the appropriate file location:

- *web_agent_home*/affwebservices/WEB-INF
- *sps_home*/secure-proxy/Tomcat/webapps/affwebservices/WEB-INF

Chapter 24: Troubleshooting Legacy Federation

This section contains the following topics:

Transaction IDs to Aid Federation Troubleshooting

Troubleshooting a federated transaction is difficult when many transactions are logged in one file. To follow a single transaction in a trace log, use the SAML transaction ID. When a federation call occurs, the FWS application first generates a SAML Transaction ID. The SAML Transaction ID is generated only once. This unique SAML transaction ID can map to multiple transaction IDs

For example, you can see the following message in the fwstrace.log for a SAML 2.0 POST transaction. Note the line in bold that shows the mapping of the two transaction IDs.

```
[08/01/2013][17:33:54][2292][1884][1c2d7650-b006e46a-ed071f41-bbbe
de33-fe78e2dd-38d][SSO.java][processAuthentication]
```
`[SAMLTransactionID 2aaf90ec-fdef4897-0ef49d91-63d4031d-f508a3e9-12 maps to TransactionID: 1c2d7650-b006e46a-ed071f41-bbbede33-fe78e2dd-38d.]`

The CA SiteMinder® Federation system generates a new SAMLTransactionID only if it is acting as the asserting party. These specific activities are:

- When Federation Web Services redirects the browser to the authentication URL to establish a session.
- For the following HTTP-Artifact single sign-on transactions:
  - When the asserting party sends the artifact to the relying party.
  - When the asserting party resolves the artifact.
- When the user is redirected to the Identity Discovery profile URL.
- During single logout at the asserting party.

At the relying party, there exists a request ID, which can be traced easily through the log files. The request ID makes it unnecessary for the CA SiteMinder® Federation system to generate a SAMLTransactionID at the relying party.

For each unique SAML transaction ID, there can be multiple transaction IDs. When a new HTTP transaction occurs, a new transaction ID is generated. This transaction ID is mapped to the single SAML transaction ID. For example, in the trace log you can see the following entries:

```
SamlTransactionID ["xyz"] maps to TransationID["123"]
["123"] HTTP operation
["123"] HTTP operation
```

A new transaction ID "456" is generated:

```
SamlTransactionID["xyz"] Maps to Transactionid["456"]
["456"] <some operation>
["456"] <some operation>
```

Transaction IDs are placed in the fwstrace.log and the smtracedefault.log. The same set of transaction IDs for a single transaction is written to each of these logs. The trail of IDs in these logs enables you to follow a transaction. If there is a failure, the IDs help you determine which event failed for a transaction.

General Issues

The following troubleshooting topics apply to SAML 1.x and SAML 2.0.

Web Agent Option Pack Fails to Initialize Due to Invalid smjavaagent.dll

Symptom:

The Web Agent Option Pack fails to initialize with on a system with other CA products. Error messages, such as "Java Agent API initialization FAILED" or "unsatisfied link error" display.

Error messages similar to the following appear in the Federation Web Service log file:

```
11:04:46 AM[29959477:E] Exception while reading the WebAgent configuration
information: javaagent_api_getConfig
11:04:46 AM[29959477:E] Java Agent API initialization FAILED.
```

Solution:

An invalid version of smjavaagentapi.dll can be present the system path. Verify that all installed products are compatible with one another and of compatible versions.

To verify the versions

1. Log in to the Technical Support site.
2. Search for the Platform Support Matrix for 12.52 SP1.

Cookie Domain Mismatch Errors

Symptom:

After successful SAML authentication at consumer/SP site, the consumer/SP Web Agent still challenges the user because of cookie domain mismatch.

Solution:

Verify that the producer/IdP and consumer/SP are not in the same cookie domain. Legacy federation does not support federation within the same cookie domain. Separate cookie domains are required at the producer/IdP and consumer/SP sites. Additionally, verify that the CookieDomainScope parameter is set to the appropriate value for your environment. This parameter is a Web Agent parameter (see information about single sign-on in the *CA SiteMinder® Web Agent Configuration Guide*.

If separate cookie domains are in use, verify that the cookie domain in the Agent configuration matches the domain name in the requested target URL.

Error After Successful Authentication at Consumer/SP

Symptom:

After successful authentication at the consumer site, an HTTP 404 "Page Not Found" error code is returned to the browser.

Solution:

Verify that the target page exists in the web server document root. Examine the FWS trace log to verify that the user is being redirected to the correct URL.

HTTP 404 Error When Trying to Retrieve Assertion at the Consumer

Symptom:

When the relying party tries to retrieve an assertion, an HTTP 404 "Page Not Found" error code is returned to the browser.

Solution:

Verify that the Federation Web Services application is deployed as a web application. Deploy the application on a web server running one of the supported application servers. The CA SiteMinder® Platform Support Matrix lists the supported platforms for the Web Agent Option Pack.

Federation Web Services Fails to Send SAML Request to Producer/IdP

Symptom:

The Federation Web Services application at the consumer/SP fails to send a SAML request message to the producer/IdP. The consuming side fails to trust the certificate of the web server.

Solution:

Add the certificate of the Certificate Authority that issued the client certificate to the key database of the web server at the producer/IdP.

Matching Parameter Case-Sensitivity Configuration Issues

Symptom:

Problems occur due to conflicts between configuration parameters that must correspond on producer/Identity Provider and consumer/Service Provider, even though the parameters appear to match.

Solution:

The URL string that comes after the colon is case-sensitive. For example, the text after **http:** is case-sensitive. Therefore, the case of the URLs in all corresponding settings must match.

Parameter values that must match between the asserting and relying parties are documented in the topic Configuration Settings that Must Use the Same Values (see page 361).

Policy Server System Fails After Logoff

Symptom:

In some environments, logging off the Policy Server while it is running causes the Policy Server to fail. The failure is due to a JVM issue.

Solution:

Add the -Xrs command to its own command line in the JVMOptions.txt file. This command is case-sensitive, so add it as shown. This command reduces usage of operating system signals by the JVM.

The JVMOptions.txt file is located in *policy_server_home*/config/.

Multibyte Characters in Assertions are Not Handled Properly

Symptom:

When you include a multibyte character in an assertion, problems can occur.

Solution:

Set the LANG setting for your operating system to UTF-8, as follows:

LANG=xx_xx.UTF-8

For example, for Japanese, the entry would be:

LANG=ja_JP.UTF-8

Trace Logs Not Appearing for IIS Web Server Using ServletExec

Symptom:

You have enabled trace logging in the LoggerConfig.properties file, but the affwebservices.log and FWStrace.log files are not being written to the WEB-INF/classes directory.

Solution:

Verifies that the anonymous user account associated with ServletExec has permissions to write to the Windows file system. If the user account does not have the right to act as part of the operating system, ServletExec cannot write the log files.

Error During Initialization of JVM

Symptom:

If you receive the following error message in the Policy Server log (figure out which log):

```
Error occurred during initialization of JVM
Could not reserve enough space for object heap.
```

The Web Agent Option Pack functionality is not working due to a JVM initialization failure.

Solution:

Restrict the object heap memory size.

To restrict the memory size

1. Open the JVMOptions.txt file, in the directory *web_agent_home*/WEB-INF/properties file.
2. Add the following entry to the file as it is written here:

   ```
   -Xms128M
   ```

3. Save the file.
4. Restart the Policy Server.

SAML 1.x-Only Issues

The following issues apply only to SAML 1.x features.

SAML 1.x Artifact Profile Single Sign-On Failing

Symptom:

If single sign-on with the SAML 1.x artifact profile is configured, the consumer site fails to send SAML request messages to the producer. Error messages similar to the following appear in the Federation Web Service log file:

```
May 23, 2012 4:20:44.234 PM[28349544:E] Dispatcher object thrown unknown exception while processing the request message. Message: java.net.ConnectException: Connection refused: connect.
May 23, 2012 4:20:44.234 PM[28349544:E] Exception caught. Message: com.netegrity.affiliateminder.webservices.m: Exception occurred while message dispatcher(srca) object trying to send SOAP request message to the SAML producer.
```

Solution:

Verify that the web server hosting the Assertion Retrieval Service is running with a configured SSL port.

Failed Authentication for Access to Assertion Retrieval Service

Symptom:

In an environment using SAML 1.x artifact single sign-on, the consumer fails authentication when trying to access the Assertion Retrieval Service at the producer.

Solution:

If basic authentication protects the Assertion Retrieval Service, verify the Name and Password for the Affiliate configuration match the Affiliate Name and Password for the SAML Artifact authentication scheme.

Authentication Fails After Modifying Authentication Method

Symptom:

If you change the authentication method protecting the SAML 1.x Assertion Retrieval Service from Basic to Client Cert, subsequent authentication requests can fail.

If you change the authentication method protecting the SAML 1.x Assertion Retrieval Service from Client Cert to Basic, subsequent authentication requests can fail.

Solution:

Restart the web server after the authentication method is changed.

Client Authentication Fails for SAML Artifact Single Sign-on

Symptom:

Client certificate authentication for SAML 1.x artifact single sign-on fails at the producer. The following error is logged in the web agent trace logs:

```
Setting HTTP response variable HTTP_consumer_name=from SiteMinder
```

For example, if the Attribute Name in the response is configured as "name" for an LDAP User Directory, the response fails.

Solution:

Verify that you create a Web Agent response under the domain FederationWebServicesDomain. The response must be as follows:

Attribute type

WebAgent HTTP Header variable

Attribute Kind

User Attribute

Variable Name

consumer_name

Attribute Name

uid (for LDAP) or name (for ODBC)

SAML 2.0-Only Issues

The following issues apply only to SAML 2.0 features.

Failed Authentication to Access the Assertion Retrieval Service

Symptom:

If you configure SAML 2.0 artifact single sign-on, the Service Provider fails to authenticate when accessing the Artifact Resolution Service at the Identity Provider.

Error messages similar to the following appear in the Federation Web Service log file:

```
May 23, 2005 4:43:51.479 PM[31538514:E] SAML producer returned error http status code.
HTTP return status: 401. Message: <HTML><HEAD><TITLE>401: Access
Denied</TITLE></HEAD><BODY><H1>401: Access Denied</H1>
Proper authorization is required for this area. Either your browser does not perform
authorization, or your authorization has failed.</BODY></HTML>
```

Solution:

Depends upon the configured authentication:

- For basic authentication, verify that the Name and Password for the SAML Service Provider matches the Affiliate Name and Password for the SAML 2.0 authentication scheme.
- For client certificate authentication to protect the Artifact Resolution Service, verify that the client certificate for the Service Provider is valid. Also, verify that the certificate is in the certificate data store. Additionally, verify that the Certificate Authority that issued the client certificate is in the certificate data store at the Identity Provider.
- If no authentication is configured, confirm the Artifact Resolution Service URL is unprotected.

ODBC Errors Deleting Expiry Data From Session Store

Symptom:

If you upgrade a Policy Server from an earlier version, ODBC errors can occur when deleting expiry data from the session store.

Solution:

Upgrade the session store schema as described in the *CA SiteMinder® Upgrade Guide*.

Appendix A: Recreate a Legacy Configuration in the Partnership Model

No direct migration path from legacy federation to [set the pfr variable for your book] exists. Reproducing your legacy federation configuration in the [set the pfr variable for your book] model requires recreating the legacy entities and configuring partnerships.

Legacy and partnership objects do not share a one-to-one correspondence. In the legacy federation model, configuring federation involves the following tasks at each partner:

Asserting party

- Configuring affiliate domains.
- Identifying the relying parties in the affiliate domains and configuring the communication with those relying parties. The relying parties include SAML 1.x affiliates, SAML 2.0 Service Providers, and WSFED Resource Partners.

Relying party

- Configuring authentication schemes that define the relying party.
- Within the authentication scheme, specifying how the relying party consumes an assertion and how the relying party redirects users to the target application.

In a partnership model, recreating a legacy configuration involves:

- Configuring asserting and relying party entities that represent the business partners.
- Defining partnerships between the entities.

The following tables shows the relationship between legacy federation components and [set the pfr variable for your book] components.

Legacy Components (Asserting Party)	Partnership Components (Asserting Party)
SAML 1.1 Affiliate	SAML 1.1 Producer-to-Consumer partnership *[set the pfr variable for your book] does not support SAML 1.0.*
SAML 2.0 Service Provider	SAML2 IdP-to-SP partnership
WSFED Resource Partner	WSFED IP-to-RP partnership

Legacy Components (Relying Party)	Partnership Components (Relying Party)
Authentication Scheme: SAML Artifact or POST Template	SAML 1.1 Consumer-to-Producer partnership
Authentication Scheme: SAML 2.0 Template	SAML2 SP-to-IdP partnership
Authentication Scheme: WS-Federation Template	WSFED RP-to-IP partnership

If you plan to recreate your legacy federation objects in the partnership model, pay attention to the following settings:

Active

(Affiliate/Service Provider Properties and SAML authentication scheme dialog for legacy federation). If you use the legacy federation configuration, confirm that this check box is selected. If you recreate the legacy configuration in the partnership federation model with similar values for identity settings, such as source ID, clear this check box before activating the partnership federation object.

CA SiteMinder® cannot work with a legacy and partnership configuration that use the same identity values or a name collision occurs.

Artifact Protection Type

(SSO settings for [set the pfr variable for your book]). Defines how the back channel is protected for HTTP-Artifact single sign-on.

If you recreate your legacy federation configuration in the [set the pfr variable for your book] model, use the legacy method of protecting the back channel. The legacy option lets the configuration use the existing URL for the Assertion Retrieval Service (SAML 1.x) or Artifact Resolution Service (SAML 2.0).

By selecting legacy as the option, CA SiteMinder® accepts the request. You do not have to modify the URL. If the artifact service URL is from the legacy configuration but only the partnership option is selected for this setting, CA SiteMinder® rejects the request.

Important! For the legacy federation option, enforce the policy that protects the artifact service. The artifact service is a component of the Federation Web Services. The software creates policies for Federation Web Services automatically. However, you are required to indicate which partnership is permitted access to the service that retrieves artifacts. For more information, refer to the *Partnership Federation Guide.*

Options: Legacy, Partnership

Note: CA SiteMinder® 12.52 SP1 ships with the Federation Security Services User Interface (FSS UI) and the Administrative UI. If you switch from the FSS UI to the Administrative UI for configuration, do not return to the FSS UI for any modifications to any configuration objects. Once you begin with the Administrative UI, continue to use the Administrative UI exclusively. If you return to the FSS UI after using the Administrative UI, objects in the policy store can impair the function of the Policy Server.