CA SiteMinder Federation Standalone

Federation Standalone Guide

r12.52

This Documentation, which includes embedded help systems and electronically distributed materials, (hereinafter referred to as the “Documentation”) is for your informational purposes only and is subject to change or withdrawal by CA at any time. This Documentation is proprietary information of CA and may not be copied, transferred, reproduced, disclosed, modified or duplicated, in whole or in part, without the prior written consent of CA.

If you are a licensed user of the software product(s) addressed in the Documentation, you may print or otherwise make available a reasonable number of copies of the Documentation for internal use by you and your employees in connection with that software, provided that all CA copyright notices and legends are affixed to each reproduced copy.

The right to print or otherwise make available copies of the Documentation is limited to the period during which the applicable license for such software remains in full force and effect. Should the license terminate for any reason, it is your responsibility to certify in writing to CA that all copies and partial copies of the Documentation have been returned to CA or destroyed.

TO THE EXTENT PERMITTED BY APPLICABLE LAW, CA PROVIDES THIS DOCUMENTATION “AS IS” WITHOUT WARRANTY OF ANY KIND, INCLUDING WITHOUT LIMITATION, ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NONINFRINGEMENT. IN NO EVENT WILL CA BE LIABLE TO YOU OR ANY THIRD PARTY FOR ANY LOSS OR DAMAGE, DIRECT OR INDIRECT, FROM THE USE OF THIS DOCUMENTATION, INCLUDING WITHOUT LIMITATION, LOST PROFITS, LOST INVESTMENT, BUSINESS INTERRUPTION, GOODWILL, OR LOST DATA, EVEN IF CA IS EXPRESSLY ADVISED IN ADVANCE OF THE POSSIBILITY OF SUCH LOSS OR DAMAGE.

The use of any software product referenced in the Documentation is governed by the applicable license agreement and such license agreement is not modified in any way by the terms of this notice.

The manufacturer of this Documentation is CA.

Provided with “Restricted Rights.” Use, duplication or disclosure by the United States Government is subject to the restrictions set forth in FAR Sections 12.212, 52.227-14, and 52.227-19(c)(1) - (2) and DFARS Section 252.227-7014(b)(3), as applicable, or their successors.

Copyright © 2013 CA. All rights reserved. All trademarks, trade names, service marks, and logos referenced herein belong to their respective companies.

CA Technologies Product References

This document references the following CA Technologies products:

- CA SiteMinder®

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

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

- Resolving Signature Verification Failures (see page 401)—Added information about disabling XML signature verification attacks in case of a signature verification failure. Resolves CQ: 168095 (STAR issue: 21321479;1).
- Logs to Monitor Federation Activities (see page 373)—Information about logging has been updated to reflect the use of log4j for the server.log file. Also, all logging information has been consolidated in the section. Resolves CQ 171956 (STAR issue 21454409-1 and CQ 165412 (Star issue: 21257428-1).
- Import a Key/Certificate Pair from an Existing File (see page 120)—Added material describing the action of the Use as CA option. Resolves CQ 173083.
- FEDSESSION cookie timeout settings (see page 181)—The Idle and Max Timeout settings have been corrected. Resolves CQ 173107.
- Back up an Existing Configuration (see page 396)—The XPSExport command option **-xa** is obsolete. The command is replaced by the command **-xe -xp**. ResolvesCQ 173659 and STAR issue 21480783-2.
- Query string method for delegated authentication (see page 228)—Added a note indicating that the query string method is for use only in a production environment. Resolves CQ 165470 and CQ 165473.
- How to Connect to an LDAP User Directory over SSL (see page 76)—Updated references from cert7.db files to cert8.db files. Resolves CQ 172315 (STAR issue: 21454358-01).
- Load Balancing (see page 337) and Failover Support (see page 325)—Updated the graphics and clarified the process and setup for these features. Resolves CQ 145146 (STAR issue: 20533073;1).

Contents

Chapter 1: Introduction 17

Overview 17
Product Components 18
FIPS 140-2 Support Offered by CA SiteMinder® Federation Standalone 19
Programmerless Federation 20
Intended Audience 21
Terminology Used in this Guide 21
Federation in Your Enterprise 23
- User Identification Across the Partnership 25
- Attributes for Customizing an Application 28
- Federation Profile for Single Sign-on 29
- The Partnership Model 29

Chapter 2: Administrative UI 31

Administrative UI Overview 31
Object Management 32
- New Object Creation 32
- Objects Lists 32
- Action Button for Object Lists 32
- Filtering Object Lists 33
- Page Displays 34

Wizards for Configuring Objects 34
Log in to the Administrative UI 34
- UI Login Password Conditions with Active Directory 35

Chapter 3: Getting Started with a Simple Partnership 37

Basic SAML 2.0 Partnership 37
Sample Federation Network 38
Configure the IdP Partner 39
- Establish a User Directory Connection 40
- Configure the Partnership Entities 42
- Create the IdP-to-SP Partnership 44
- Specify Federation Users for Assertion Generation 45
- Add a Name ID to the Assertion 45
- Set Up Single Sign-on 46
- Disable Signature Processing 46

Confirm the IdP-to-SP Partnership Settings 47
Configure the SP Partner 47
Establish a User Directory Connection 47
Identify the Partnership Entities 50
Create the SP-to-IdP Partnership 52
Specify the User Identification Attribute 53
Configure Single Sign-on 53
Disable Signature Processing 54
Confirm the SP Partner Settings 54
Activate the Partnership 54
Test the Partnership (POST Profile) 55
Create a Web Page to Initiate Single Sign-on 55
Create a Target Resource 55
Test POST Single Sign-on 56
Enable Signature Processing 56
Configure Signature Processing at the IdP 57
Configure Signature Processing at the SP 59
Add Single Logout 60
Configure Single Logout at the IdP 61
Configure Single Logout at the SP 62
Test Single Logout 63
Set Up the Artifact Profile for SSO 63
Configure Artifact SSO at the IdP 64
Configure Artifact SSO at the SP 64
Test the Partnership (Artifact SSO) 65
Configuration Procedures Beyond the Simple Partnership 67

Chapter 4: Storing User Session, Assertion, and Expiry Data 69

Federation Data Stored in the Session Store 69
Enable the Session Store 70
Environments that Require a Shared Session Store 71

Chapter 5: User Directory Connections for Authentication 73

User Directory Management Overview 73
LDAP Directory Connection 74
Load Balancing and Failover for LDAP User Directories 74
How to Connect to an LDAP User Directory Over SSL 76
Before You Configure an LDAP Connection over SSL 76
Create the Certificate Database Files 77
Add the Root Certificate Authority to the Certificate Database 78
Add the Server Certificate to the Certificate Database 80

Verify that the Certificates are in the Database 81
SSL-enable the LDAP User Directory Connection 82
Establish a Connection to the Certificate Database 83
Verify the SSL Connection to the LDAP Directory 84
Troubleshoot the SSL Connection to the LDAP User Directory 84
ODBC Directory Connection 85
ODBC Directory Failover Configuration 85
ODBC Data Source on Solaris Configuration Requirement 87
Test a User Directory Connection from the Directory List 89
Create a Common View of the Same User Information Across Directories 89
Establish Connections to User Directories 90
Configure User Attribute Mappings 91
Apply Mappings to Assertion Attributes 105

Chapter 6: Federation Entity Configuration 107

Methods to Create an Entity 107
How to Create an Entity without Using Metadata 107
Entity Type Choice 107
Detailed Local Entity Configuration 108
Detailed Remote Entity Configuration 109
Confirm the Entity Configuration 110
Entity Configuration Changes from a Partnership 111
How to Create an Entity by Importing Metadata 111
Metadata File Selection 112
Select an Entity to Import 113
Certificate Imports 113
Confirm the Entity Configuration 114

Chapter 7: Key and Certificate Management 115

Certificate and Private Key Usage 115
Aliases to Reference Certificate Data Store Content 116
Signing and Verification Operations 117
Encryption and Decryption Operations 118
Certificates for SSL Connections 118
Certificates to Secure the Artifact Back Channel 118
Obtain a Key/Certificate Pair for Federated Transactions 119
Import a Key/Certificate Pair from an Existing File 120
How to Generate a Key/Certificate Pair 122
Generate a New Certificate Signing Request 124
How to Verify that Certificates are Valid Using CRLs 124
Add a CRL to the CDS 126

Update a CRL 127
Manage Certificate Cache Refresh and Grace Period 128
How to Verify that Certificates are Valid using OCSP 129
OCSP Prerequisites 130
Add an OCSP Responder to the CDS 130
Enable OCSP Status Checks 131
Manage Certificate Cache Refresh and Grace Period 131
How to Send Certificates to Your Partner 132
Generate a New Key/Certificate Pair Using the UI or a Third-party Tool 134
Import the Key/Cert Pair into the CDS 135
Export Certificates from the CDS using the Administrative UI 137
Send the Certificate File to your Partner 138
Update Certificates in the Certificate Data Store 138
Certificate Authority (CA) Certificate Usage 139
Import a CA Certificate 139
Troubleshoot Certificate Signature Verification for Back Channel Communication 141

Chapter 8: Partnership Creation and Activation 143

Partnership Creation 143
Partnership Definition 144
Partnership Identification and Configuration 145
Editing Entities from the Partnership 146
Partnership Confirmation 146
Partnership Activation 147
Exporting a Partnership 147

Chapter 9: Federated User Identification for a Partnership 149

Federation Users Configuration at the Asserting Party 149
Configure Federation Users 150
User Identification at the Relying Party 153
Configure User Identification at the Relying Party 155
Employ AllowCreate for User Identification (SAML 2.0) 156

Chapter 10: Assertion Configuration at the Asserting Party 157

Assertion Configuration 157
Configure Assertion Options 158
Assertion Attribute Configuration Examples 160
How To Add Session Attributes to an Assertion 160
Determine which Session Attributes are Available 161
Add Session Attributes to the Assertion Configuration 161

Confirm the Authentication Mode and URL for SSO 163
How to Configure Claims Transformation at the Asserting Party 163
Prerequisites for Claims Transformation 165
Learn the Attribute Expression Guidelines 165
Configure Claims Transformation at the Asserting Party 167
Customize Assertion Content 173
Implement the AssertionGeneratorPlugin Interface 174
Deploy an Assertion Generator Plug-in 174
Enable the Assertion Generator Plug-in 175

Chapter 11: Assertion Processing Customization (Relying Party) 177

Customize Assertion Processing (Relying Party) 177
Implement the MessageConsumerPlugin Interface 178
Enable the Message Consumer Plug-in in the UI 179
Deploy a Message Consumer Plug-in 180

Chapter 12: Single Sign-on Configuration 181

Single Sign-on Configuration (Asserting Party) 181
Customize the Auto-POST form for HTTP-POST SSO 184
Authentication Options using Partnership Federation 185
Single Sign-on Configuration (Relying Party) 185
Assertion Validity for Single Sign-on 186
Session Validity at a Service Provider 188
Status Redirects for HTTP Errors (SAML 2.0 IdP) 189
Single Sign-on Initiation (SAML 2.0) 189
Back Channel Authentication for Artifact SSO 190
Configure the HTTP-Artifact Back Channel 191
How to Enable SAML 2.0 Attribute Query Support 192
Configure the Partnership for Attribute Query Support 194
Configure the SAML 2.0 Attribute Authority 194
How to Retrieve User Attribute Values from a Third-Party Source 195
Proxied Attribute Query Overview 196
Enable the System to Serve as an Attribute Authority (IdP->SP) 197
Enable the System to Serve as an Attribute Requester (SP->IdP) 198
How to Get User Consent to Send an Assertion 199
User Consent Example 201
Enable User Consent at the IdP 201
Customize a User Consent Form (Optional) 202
Require User Consent at the SP 203
Enhanced Client or Proxy Profile Overview (SAML 2.0) 203
Configure ECP at the Identity Provider 205

Configure ECP at the Service Provider 205
IDP Discovery Profile (SAML 2.0) 206
IDP Discovery Configuration at the Identity Provider 206
IDP Discovery Configuration at the Service Provider 207
Securing the IdP Discovery Target Against Attacks 208
How to Configure SAML 2.0 HTTP POST Binding 209
Enable the HTTP POST Binding at the IdP 210
Enable the HTTP POST Binding at the SP 211

Chapter 13: Configure Social Sign-on 213

Authenticate Users Using an OAuth Authorization Server 214
Verify the Prerequisites 216
Create a Local OAuth Client Entity 216
Create or Modify the Remote Entity of an Authorization Server 217
Create a Partnership to Configure Single Sign-On 218
Migrate an OAuth Authentication Scheme Set-up to OAuth Partnership 219
Configure a Credential Selector Page 220
Configure Single Sign-On Between the Federation System and an Identity Provider 223
Create an Authentication Method Group 223
Configure a Partnership Between the Federation System and Enterprise 224
Customize the Header and Footer on the Credential Selector Page 225

Chapter 14: Delegated Authentication 227

Delegated Authentication Overview 227
How the Third Party WAM Passes the User Identity 228
Cookie Method for Passing User Identity 229
Query String Method for Passing User Identity 231
Delegated Authentication Configuration 234
Cookie Delegated Authentication Sample Setup 234
Query String Delegated Authentication Sample Setup 235
Third-party WAM Configuration for Cookie Delegated Authentication 237
Third-party WAM Configuration for Query String Delegated Authentication 238

Chapter 15: URLs to Initiate Single Sign-on 239

Links to Servlets which Initiate Single Sign-on 239
Producer-initiated SSO (SAML 1.1) 239
IdP-initiated SSO (SAML 2.0 Artifact or POST) 241
Unsolicited Response Query Parameters Used by the IdP 242
ForceAuthn and IsPassive Processing at the IdP 243
SP-initiated SSO (SAML 2.0) 244

AuthnRequest Query Parameters Used by an SP 246
IP-initiated Single Sign-on (WSFED) 248
RP-initiated Single Sign-on (WSFED) 249

Chapter 16: Logging Out of User Sessions 251

Single Logout (SAML 2.0) 251
Managing Single Logout Across a Network Using HTTP-Redirect and SOAP 252
Understanding Skew Time for SLO Request Validity 253
Configure Single Logout 253
Back Channel Configuration for Single Logout 255
Sign-Out Overview (WS-Federation) 257
Enable WSFED Sign-Out 257
Local Logout at the SP (SAML 2.0) 258

Chapter 17: Authentication Context Processing (SAML 2.0) 259

Determine how a User Authenticated at an Identity Provider 260
Authentication Context Processing for IdP-initiated SSO 262
Authentication Context Processing for SP-Initiated SSO 262
Determine Authentication Context and Strength Levels with your Partner 264
Configure an Authentication Context Template 264
Set up an Authentication Context Template 265
Protection Level Assignments for a Context Template 266
Configure Authentication Context Processing at the IdP 268
Specify How the IdP Obtains the Authentication Context 268
Configure Authentication Context Requests at the SP 270
Enable Authentication Context Requests at the SP 271

Chapter 18: Sign and Encrypt Federation Messages 273

Signature Configuration at a SAML 1.1 Producer and WSFED IP 273
Signature Verification at a SAML 1.1 Consumer and a WSFED RP 274
Signature Configuration at a SAML 2.0 IdP 275
Encryption Configuration at a SAML 2.0 IdP 276
Signature Configuration at a SAML 2.0 SP 277
Encryption Configuration at a SAML 2.0 SP 279

Chapter 19: Session Duration Management at a Service Provider 281

How to Manage the Authentication Session Duration at a Service Provider 281
Include a Session Duration Attribute in an Assertion 282

Chapter 20: CA SiteMinder® Integration with CA SiteMinder® Federation Standalone 285

How to Integrate CA SiteMinder® Federation Standalone and CA SiteMinder® 285
Integrate with CA SiteMinder® using the SiteMinder Connector 287
Configure a Policy to Generate a Session at Each Site 289
Configure the Connector Settings 292
Enable the Connector at the Partnership Level 293

Chapter 21: Secure a Federated Environment 295

Protecting Federated Communication 295
Enforcing the One Time Use of an Assertion 295
Securing Connections Across the Federated Environment 296
Protecting a Federated Network Against Cross-Site Scripting 296

Chapter 22: Application Integration at the Relying Party 299

Relying Party Interaction with Applications 299
Redirect a User to the Target Application 299
Using HTTP Headers to Pass Assertion Data (SAML only) 301
Configure HTTP Headers to Pass Assertion Data (SAML only) 302
Mapping Assertion Attributes to Application Attributes (SAML Only) 302
Using the Application Attributes Definitions Table 303
Modify and Delete Mappings 304
Construct Attribute Mapping Rules Using the Proper Syntax 305
Configure Attribute Mapping at the Relying Party 307
Dynamic Provisioning of a User Identity at the Relying Party 308
Local Account Linking for Provisioning 309
Remote Provisioning 312
Delivery of Assertion Data to the Provisioning Application 313
Remote Provisioning Configuration 315
Failed Authentication Handling Using Redirect URLs (Relying Party) 316

Chapter 23: Export Metadata to Aid Partnership Configuration 319

Metadata Export Overview 319
Entity-level Metadata Export 320
Partnership-Level Metadata Export 320
How To Enable WS-Federation Metadata Exchange 321
Metadata Exchange Transaction Flow 322
Give the Metadata Exchange URL to Your Partner 323
Enable WSFED Metadata Exchange 323

Chapter 24: Failover Support for Federation System 325

Failover Introduction 325
How to Configure Failover 327
Set up Failover at Each Federation System 327
Set up the Proxy Server or Load Balancer for Failover 329
How to Configure Failover with SSL Enabled 329
Configure SSL-enabled Failover Behind a Load Balancer 330
Configure SSL-enabled Failover Behind a Proxy Server 333
Set up the Proxy Server or Load Balancer for Failover 334
Maintain the Same Configuration for Each System 334

Chapter 25: Load Balancing Support for Federation System 337

How to Configure Load Balancing 337
Configure the Load Balancer 340
Set up the Federation Systems to Work with a Load Balancer 341
Configure Redirections to an SSL Load Balancer (optional) 343

Chapter 26: Federation System Administration 345

Server Status Monitoring 345
Modify the System Settings 345
Deployment Settings 346
Deployment Modes and FIPS Settings 346
HTTP Header Protection for a Proxy Mode Deployment at the Relying Party 347
CA SiteMinder® Connector Settings 348
Cookie Settings for Session and Identity Cookies 350
How to Configure Federation System Administrators 351
Connect to External User Stores 352
Select Users as Administrators 353
Change the Default Administrator Password (Optional) 354
Administrator Session Management 355
Administrative Session Interaction 356
Disable UI Administration 357

Chapter 27: SSL Administration for Federation System 359

SSL Administration for the Apache Web Server and the UI 359
How to Enable SSL for the Apache Web Server and the UI 360
Deactivate SSL 363
Reactivate SSL 365
Replace or Resubmit a Certificate Signing Request for SSL 366

Remove SSL from the Embedded Apache Server and the UI 366
How to Migrate SSL Keys and Certificates 367
Copy Key and Certificate Files from the r12 System 369
Copy the SSL Migration Tool to Same Folder as the Key/Certificate Files 369
Migrate or Export SSL Keys and Certificates 369
SSL Migration Tool Command Arguments 370

Chapter 28: Logs to Monitor Federation Activities 373

Federation Logging Overview 373
Federation Web Services (FWS) Logging 374
Server Trace Logging 376
Set Up the Server Trace Log Configuration File 376
Configure the Behavior of the Server Trace Log File 377
server.log File Setup 379
Log Settings 381
log4j.properties File for server.log 383
Federation Data Object Trace Logging 384
Audit Logging 385
Set the Audit Log Name and Location (Optional) 386
Use an ODBC Database for Audit Logging (Optional) 387
Transaction IDs to Aid Federation Troubleshooting 392
How To Follow a Single Transaction in a Log 393

Chapter 29: Restore Federation System Configuration 395

How To Restore a System to a Previous Configuration 395
Back up an Existing Configuration 396
Revert to a Backed-up Configuration 396

Chapter 30: Troubleshooting 399

System Performance Troubleshooting 399
Configure the Session Store Timeout for Heavy Load Conditions 399
Proxy Engine Hangs and Stops Processing Requests 400
Resolving Signature Verification Failures 401
Two SSO Transactions Fail Using the Same Browser Session 402
Examine Secure Proxy Engine Logs to Troubleshoot the System 402

Appendix A: Open Format Cookie Details 405

Contents of the Open Format Cookie 405

Appendix B: Encryption and Decryption Algorithms 409

Open Format Cookie Encryption Algorithms....409
Digital Signing and Private Key Algorithms....410
Back Channel Communication Algorithms....410
Backend Communication Algorithms (SPS Server)....411
Java SDK Encryption Algorithms....411
Federation System Crypto Algorithm....411
Internal Key Encryption Algorithms....412
SSL Key Algorithms for the Apache Web Server and Administrative UI....412

Index 413

Chapter 1: Introduction

This section contains the following topics:

Overview (see page 17)
Product Components (see page 18)
FIPS 140-2 Support Offered by CA SiteMinder® Federation Standalone (see page 19)
Programmerless Federation (see page 20)
Intended Audience (see page 21)
Terminology Used in this Guide (see page 21)
Federation in Your Enterprise (see page 23)

Overview

Enterprise applications and services continue to become increasingly distributed across organizations. As a result, the need for secure but seamless access to services from one domain to another has increased. Federated partnerships address this need by enabling identity information to be flexible and portable, offering secure single sign-on and single logout across a network of trusted business partners.

CA SiteMinder® Federation Standalone enables customers to establish federated partnerships in a flexible way, together with or independent of a Web access management system. This federation solution lets an organization can act as the asserting party or the relying party.

Some product highlights include:

- SAML 1.1, SAML 2.0, and WS-Federation protocol support.
- Support for FIPS 140-2 compatible encryption.
- A standalone product that does not require the existence of any federation software on the target system.
- The ability to be deployed as a standalone entity or together with CA SiteMinder® Web Access Manager for federation and access control.
- The ability to pass identity data to a target application as an encrypted cookie or header.

Product Components

CA SiteMinder® Federation Standalone includes the following components:

- Secure proxy engine

  Forwards traffic to backend servers. This engine employs web server, servlet engine, proxy server and federation web services features.

  The secure proxy engine includes the following components:

  - Apache Web Server

    Acts as the HTTP listener, handling HTTP traffic for incoming requests, and can handle HTTPS traffic, once properly configured.

  - Tomcat server

    Provides a servlet container for the operation of the Administrative UI. The Apache web server communicates to the Tomcat server via a Tomcat connector called mod_jk.

- Federation server

  Enables user directory connectivity, authentication functions, and session store abilities.

- Extensible policy store

  Stores all CA SiteMinder® Federation Standalone data objects.

- Web-based user interface

  Administers the configuration of federation entities and partnerships, private keys and certificates, and various server settings.

FIPS 140-2 Support Offered by CA SiteMinder® Federation Standalone

CA SiteMinder® Federation Standalone uses certified Federal Information Processing Standard (FIPS) 140-2 compliant cryptographic libraries. These libraries provide a FIPS mode of operation when an environment uses only FIPS-compliant Advanced Encryption Standard (AES) algorithms to encrypt sensitive data.

You can install the product in one of the following FIPS modes of operation:

FIPS_COMPAT

FIPS_COMPAT (compatibility) mode is the default FIPS mode of operation during installation. In FIPS_COMPAT mode, the system continues to support the current set of non-FIPS algorithms as well as the supported FIPS-compliant algorithms.

FIPS_COMPAT mode is compatible with previous versions of the product. This compatibility enables environments with a version earlier than r12.52 to interoperate with r12.52. FIPS_COMPAT is also suitable for any clients who are satisfied with the degree of security available in the current product implementation.

If your organization does not require the use of FIPS, install the product in FIPS_COMPAT mode. No further configuration is required.

FIPS_ONLY

In FIPS_ONLY mode, the environment uses only FIPS-compliant algorithms to encrypt sensitive data.

Install the product in FIPS_ONLY mode for new installations where you want to use only FIPS-compliant algorithms.

An appendix (see page 409) in this guide lists the specific encryption and decryption algorithms that the system uses when operating in different FIPS modes.

Important! An r12.52 installation running in FIPS_ONLY mode cannot interoperate with, or be backward compatible to, earlier versions of the product, including any previous versions of APIs exposed by the product. Re-link all such software with the r12.52 versions of the respective SDKs to achieve the required support for full FIPS_ONLY mode.

Programmerless Federation

Programmerless federation is an HTTP-based approach for allowing the secure authentication, user disambiguation, inspection, and modification of SAML assertions. The advantage of programmerless federation is that applications can accomplish these tasks without having to use a language-specific SDK or other bindings.

Programmerless federation relies on HTTP/HTTPS requests and responses. These requests and responses are accessible through URLs and HTML-based protocols using web services that are an implementation of Representational State Transfer (REST) system architecture.

Any application that can issue HTTP requests, read HTTP responses, and parse XML to take advantage of the CA SiteMinder® Federation Standalone programmerless functionality.

An essential part of programmerless federation is its ability to secure the exchange of data. To secure data, CA SiteMinder® Federation Standalone uses an open format cookie. The open format cookie is a well-defined cookie format that supports strong encryption algorithms. The encrypted cookie secures the response to a request between CA SiteMinder® Federation Standalone and the local or remote applications, which can be written in any programming language that supports the same encryption and decryption algorithms that the open format cookie uses, such as Perl or Ruby.

CA SiteMinder® Federation Standalone SDKs also support the open format cookie, allowing a mix of applications.

The following CA SiteMinder® Federation Standalone features implement the programmerless federation model:

Delegated Authentication

Delegated authentication lets CA SiteMinder® Federation Standalone use a third-party web access management (WAM) system to perform the authentication of any user who requests a protected federated resource. The third-party WAM performs the authentication and then sends the federated user identity to CA SiteMinder® Federation Standalone.

Communication for delegated authentication is handled by HTTP/HTTPS requests and responses.

Provisioning at the Relying Party

Provisioning is the process of creating client accounts with the necessary account rights and access privileges for accessing data and applications. CA SiteMinder® Federation Standalone provisioning can establish a new account for a user, or populate an existing user account with information sent in a SAML assertion.

Remote provisioning is one of the provisioning methods. Remote provisioning uses an independent provisioning application to establish a user record. To pass assertion data, CA SiteMinder® Federation Standalone creates an encrypted cookie containing the data. This cookie is sent to the remote provisioning application, which is responsible for creating the user account.

Communication for provisioning is handled by HTTP/HTTPS requests and responses.

Intended Audience

This guide assumes that you understand the following concepts:

- Basic SAML fundamentals
- SAML bindings POST and artifact
- SAML profiles, such as Single Sign-on (SSO), Single logout (SLO), and Enhanced Client or Proxy (ECP)
- Public Key Infrastructure (PKI) fundamentals
- Secure Socket Layer communication basics

Terminology Used in this Guide

The following terms are used in this guide:

Asserting Party

A SAML authority that generates an assertion for use by a relying party. The asserting party creates, maintains, and manages identity information for users and provides user authentication to other relying parties. For SAML 1.1, the asserting party is known as the Producer. For SAML 2.0 and WS-Federation, the asserting party is known as the Identity Provider.

Assertion Consumer Service (SAML 1.1 and 2.0)

A Service Provider component that receives a SAML artifact or an HTTP form with an embedded SAML response and obtains the corresponding SAML assertion. The Assertion Consumer Service issues federation session cookies, and if you are integrating with CA SiteMinder®, a CA SiteMinder® session cookie.

Assertion Retrieval Service (SAML 1.1)

A Producer-side service that handles SAML 1.1 authentication using HTTP Artifact binding. This service retrieves the assertion stored at the Producer.

Artifact Resolution Service (SAML 2.0)

An Identity Provider-side service that performs SAML 2.0 authentication using the HTTP Artifact binding. This service retrieves the assertion stored at the Identity Provider.

AuthnRequest Service (SAML 2.0)

A service that enables a Service Provider to generate an AuthnRequest message for cross-domain single sign-on. This message contains information that enables CA SiteMinder® Federation Standalone to redirect the browser to the Single Sign-on Service at the Identity Provider. The AuthnRequest service is used for single sign-on using POST and artifact binding.

Note: The format of the AuthnRequest message issued by this service is specified in the Profiles for the OASIS Security Assertion Markup Language (SAML) V2.0.

Delegated Authentication

A feature that allows the use of a third-party Web Access Management system to authenticate users and then redirect the users back to CA SiteMinder® Federation Standalone to proceed with the federation process.

Legacy Cookie

(Formerly the FEDPROFILE cookie) A cookie that contains user identity information. This cookie supports only PBE encryption algorithms, which are not FIPS-compliant.

At the asserting party, a Java SDK creates the legacy cookie and CA SiteMinder® Federation Standalone reads it. At the relying party, CA SiteMinder® Federation Standalone creates the legacy cookie for use by Java-based end-user applications. The applications use the Java SDK to read the cookie.

Open Format Cookie

A cookie that contains user identity information. The open format cookie can be encrypted using FIPS or non-FIPS compatible algorithms, depending on how you generate it. You can create an open format cookie using a CA SiteMinder® Federation Standalone SDK or create it manually using any programming language that supports UTF-8 encoding.

If you require a FIPS-encrypted open format cookie, use a CA SiteMinder® Federation Standalone SDK to create the cookie and to read the cookie. The CA SiteMinder® Federation Standalone Java SDK can encrypt the cookie using a FIPS-compliant (AES) algorithm or a non-FIPS (PBE) algorithm. The CA SiteMinder® Federation Standalone .NET SDK can encrypt the cookie using only a FIPS-compatible algorithm.

Relying Party

A SAML entity that uses information from a SAML authority to provide access to services. The relying party uses assertions it receives from an asserting party to authenticate a user. For SAML 1.1, the relying party is known as the Consumer. For SAML 2.0, the relying party is known as the Service Provider.

Important! In this guide, the term *relying party* is used to mean a consumer or a Service Provider.

Single Logout Service (SAML 2.0)

This service allows a user to log out of all applications in the federation simultaneously with a single logout event. An Identity Provider or a Service Provider can initiate single logout.

Single Sign-on Service (SAML 1.1 and SAML 2.0)

For SAML 1.1, the SSO service enables a Producer to process Producer-initiated requests for federated resources.

For SAML 2.0, the SSO service enables an Identity Provider to process IdP-or SP-initiated requests for federated resources.

The Producer/IdP gathers the necessary information from the Consumer/SP to generate an assertion, which it passes back to the Consumer/SP. The Consumer/SP then uses the assertion for authentication.

Unified Expression Language

The Unified Expression Language (UEL) is a special Java expression syntax used primarily by Java web applications. You can use the UEL for embedding expressions into web pages. For CA SiteMinder® Federation Standalone, the UEL is the language you must use to define mappings between assertion attributes and application attributes at the relying party.

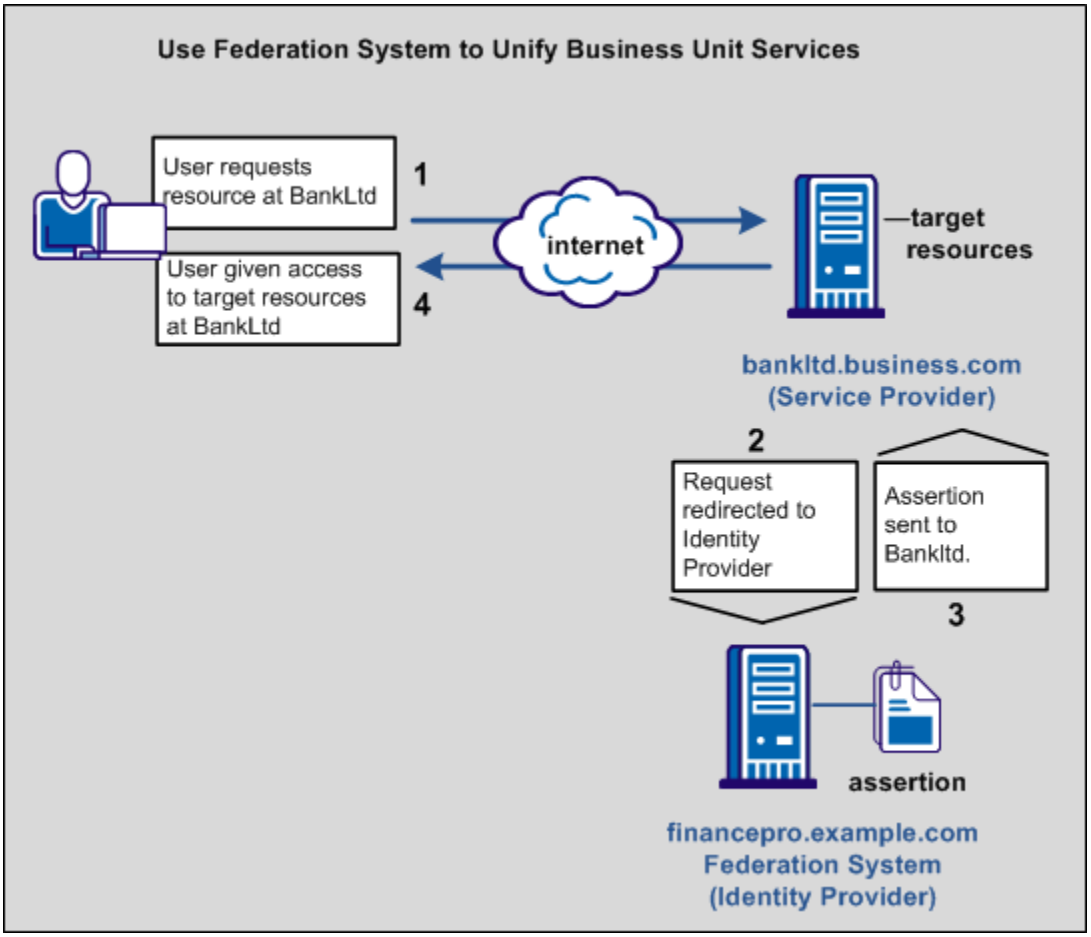
Federation in Your Enterprise

A sample business case best illustrates how the federation system can solve a common business problem.

In this business case, Financepro is a financial planning firm that recently bought the banking firm BankLtd to provide private banking to its clients. These two companies have different information infrastructures, but they want to appear as one company to their customers. To solve this problem, they set up a federated partnership.

By establishing a federated relationship, the two companies can provide a seamless customer experience using single sign-on. Customers can travel between Financepro and BankLtd without constantly being challenged to authenticate. Additionally, the sharing of customer identities and customer information can further customize the user experience and cross-promote the financial products of each partner.

The following graphic shows the federated partnership between Financepro and BankLtd. The flow of communication is based on a SAML 2.0 Service Provider-initiated single sign-on.



The illustration describes the following information flow:

1. The user tries to access a federated resource at BankLtd.
2. The user is redirected to the Financepro for authentication and the assertion is generated.
3. The assertion is passed back to BankLtd.
4. Single sign-on occurs based on either a SAML HTTP-Artifact or HTTP-POST. The user gets access to the target resource.

For this partnership to work, decide how the partnership functions before implementing the relationship using CA SiteMinder® Federation Standalone.

The issues to consider include:

- How users are identified across the partnership.
- What attributes get sent in an assertion and for what purpose.
- Which federation binding to use (SAML or WS-Federation)

Your decisions help structure the business partnership.

User Identification Across the Partnership

Business partners have their own method of defining user identity in their respective user stores. How users are identified determines how one partner can map its users to the other partner.

Consider the following scenarios:

- The User ID is the same at the user store of each site.

  Account linking is the method of user identification.

- The User ID is unique at the user store of each site.

  Identity mapping is the method of user identification. At FinancePro, a customer is identified as JohnDoe, while at BankLtd this same customer is identified as DoeJ. The partners must agree on a user attribute profile to use for identity mapping.

- The User ID does not exist at the relying party.

  Account provisioning is the method for user identification. Provisioning an account can require creating an account for a user or simply populating an existing user account with information in the SAML assertion.

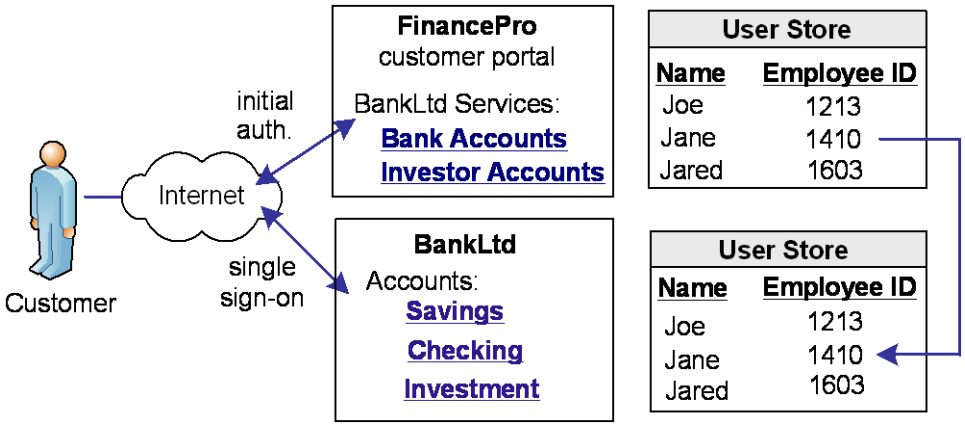
The user identification decision determines what information is sent as the user identity in the assertion.

Account Linking to Establish a Federated Identity

When a customer at FinancePro accesses a resource at BankLtd, the NameID is always in the assertion. This identifier allows BankLtd to determine who the customer is and the level of access to allow for that customer.

The NameID can establish a federated identity when the user store at each partner identifies the users in the same way with the same ID.

The following figure shows the user store at each site with the same employee IDs.



CA SiteMinder® Federation Standalone lets you configure account linking as part of the partnership configuration process. You specify a NameID format and Name ID type, which determines the type of value that defines the Name. You associate the specific Name ID type, with a static, user, or DN attribute from a user directory. The NameID that CA SiteMinder® Federation Standalone includes in the assertion conforms to the configuration you define.

When the relying party receives the assertion, the user disambiguation process at BankLtd occurs. The process links the NameID value in the assertion to a record in its user store.

Identity Mapping to Establish a Federated Identity

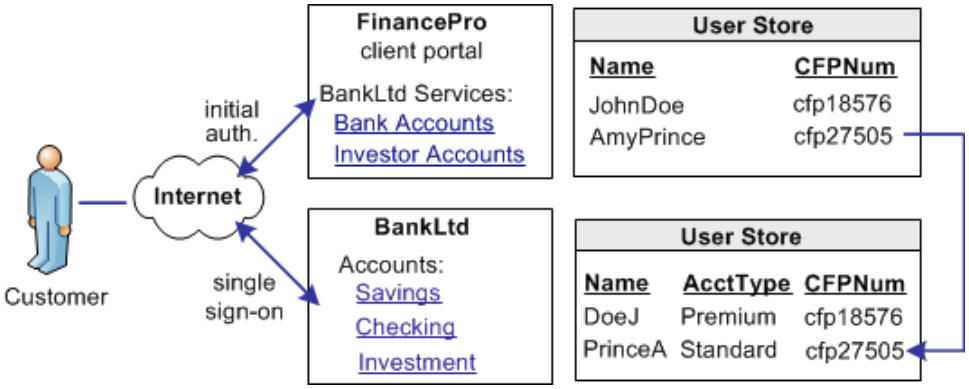
An investor at Financepro authenticates and selects a link to access information at BankLtd. The investor is taken directly to the accounts area of the BankLtd website without having to sign on.

BankLtd maintains user identities for all customers at Financepro, but the identities differ from the identities at FinancePro. For example, at FinancePro, JohnDoe is a customer. At BankLtd, this same customer is identified as DoeJ. Regardless, BankLtd must control access to sensitive portions of the company website. To establish the federated identity, the partners agree on an attribute that maps to the appropriate identity for a single customer at either site.

The partners agree on which attribute to use during an out-of-band exchange of information, meaning that the agreement is not part of any communication in any message over a channel. For this example, the attribute that the partners agree upon is a certified financial planner license number, referred to as the CFPNum in each user store.

When a customer tries accessing the federated resource at BankLtd, the request triggers the single sign-on process. The assertion that is generated at FinancePro contains the CFPNum attribute. When BankLtd receives the assertion, an application at its site has to perform the user disambiguation process. The process relies on the attribute to determine which profile identity is used for the request.

The following illustration shows how the same users are identified differently at each partner.



CA SiteMinder® Federation lets you configure identity mapping as part of the partnership configuration process. For the NameID and attribute configuration, you define an attribute called CFPID. Associate this attribute with the user attribute CFPNum, which is the name of the attribute in the user store at each partner.

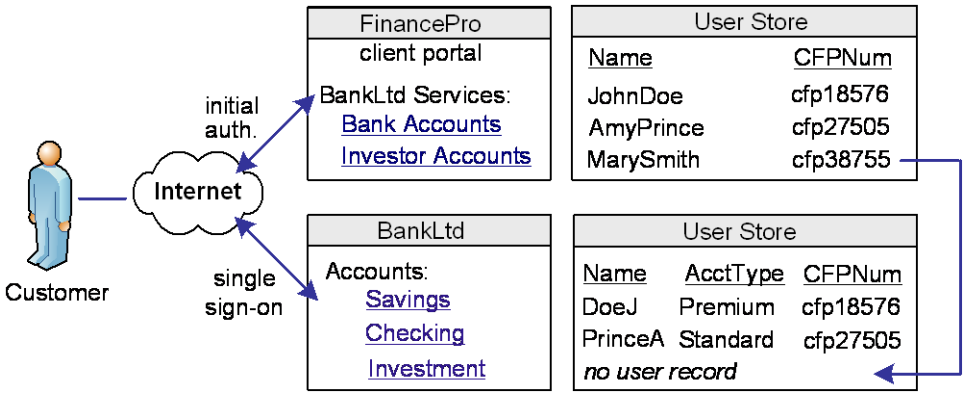
CA SiteMinder® Federation includes the attribute in the assertion. When BankLtd receives the assertion, the user disambiguation process links the attribute in the assertion to the appropriate record in its user store.

User Provisioning to Establish a Federated Identity

An investor at Financepro, Mary Smith, authenticates and clicks a link to access information at BankLtd. Initially, BankLtd cannot find a user account for Mary Smith. BankLtd wants to protect sensitive portions of its website while allowing new customers.

BankLtd has configured CA SiteMinder® Federation Standalone to implement provisioning to establish the new federated identity for Mary Smith. CA SiteMinder® Federation Standalone redirects Mary Smith to the provisioning server at BankLtd. The provisioning application, using identity information from CA SiteMinder® Federation Standalone, creates a user account in the user store.

The following illustration shows the user stores at FinancePro and BankLtd.



CA SiteMinder® Federation Standalone lets you configure provisioning as part of the partnership configuration at the relying party. In this example, you select remote provisioning and determine how assertion data is delivered to the BankLtd provisioning server. This configuration enables the dynamic creation of a user entry in the user store.

Attributes for Customizing an Application

CA SiteMinder® Federation Standalone offers two ways of using attributes to customize target applications.

Attributes Added to Assertions at the Asserting Party

You can include attributes from a user store record in an assertion to identify a user for the purpose of customizing an application.

Servlets, web applications, and other custom applications can use attributes to display customized content or enable and disable other custom features. When used with web applications, attributes can implement fine-grained access control by limiting user activity at the target site. For example, you send an attribute variable named Account Balance and set it to reflect the account holdings of the user at BankLtd.

Attributes take the form of name/value pairs. When the relying party receives the assertion, it makes the attribute values available to applications.

Attribute Mapping at the Relying Party

The relying party receives a set of assertion attributes, which can be mapped to a set of application attributes being delivered to the target application.

For example, FinancePro includes an assertion attribute CellNo=5555555555. At BankLtd, this attribute name is transformed to an application attribute Mobile=5555555555. The attribute name is converted but the value remains the same.

Multiple assertion attributes can also be transformed into a single application attribute. For example, FinancePro sends an incoming assertion with the attributes Acct=Savings and Type=Retirement and transformed at BankLtd into FundType= Retirement Savings.

More information:

Partnership Creation and Activation (see page 143)

Federation Profile for Single Sign-on

Determining the profile for a partnership depends on the binding that each side can support.

For a new federation, there are no legacy requirements for either partner. Therefore, the recommended profile to use for single sign-on is SAML 2.0 POST profile. SAML 2.0 POST profile offers secure transmission of assertion data and the configuration process is simpler than SAML Artifact profile. If, however, the agreement of two partners requires SAML Artifact, this binding can also be implemented.

The Partnership Model

The CA SiteMinder® Federation Standalone partnership model can establish a federation between Financepro and BankLtd to ease the experience of moving between the sites of each company and to verify that they appear as one company.

The Administrative UI focuses on partnership creation and identifying each side of the partnership to accomplish single sign-on.

These steps include:

1. Configuring a Partnership—Names the partnership and identifies the two entities that make up the partnership.
2. Establishing the Federation Users/User Identification—Specifies the users for which the asserting party generates assertions and the relying party authenticates.

3. NameID and Attributes—Determines how a federated identity is established and lets you add attributes to identify and customize the content of the assertion.

   Using NameID and attributes, you can verify that the appropriate information is available to the application at the relying party. This is where account linking and identity mapping would be configured.

4. SSO—Defines Single Sign-on (Artifact or POST binding), including the location of the service consuming assertions at the relying party. For SAML 2.0, additional features, such as single logout (SLO), Enhanced Client or Proxy (ECP) profile, and Identity Provider Discovery profile can be configured.

5. Signature and Encryption—Defines the signature and encryption options for secure exchange of assertions, authentication requests, and for SAML 2.0 single logout requests and responses.

6. Application Integration—Enables you to configure redirection to the target application, lets you set up provisioning of user records, and define relying-party side attribute mapping. You can also set up redirects for failed user authentication.

Chapter 2: Administrative UI

This section contains the following topics:

Administrative UI Overview (see page 31)
Object Management (see page 32)
Wizards for Configuring Objects (see page 34)
Log in to the Administrative UI (see page 34)

Administrative UI Overview

Configuration is managed through the Administrative UI. The Administrative UI is a web application that provides an administrator with access to all the system management functions of the product.

An administrator is a user who has privileges and responsibilities for managing a federated solution. Multiple administrators can be responsible for managing the system. Review the procedures for configuring multiple administrators.

The Administrative UI is organized into configuration tabs, sub-categories, lists, and task buttons:

When navigating the UI, be aware of the following:

- Main configuration categories are reflected by the following tabs:
  - Federation
  - Certs & Keys
  - User Directory
  - Infrastructure

  You navigate to one of these tabs first when configuring an object.
- Sub-categories within each main configuration tab can be selected to configure specific aspects of the federation setup.
- Lists of objects are displayed for most sub-categories. These lists contain contextual links to access existing objects and task buttons to create or modify objects.

Object Management

The Administrative UI lets you create, view, modify, and delete objects. Although the details of each task differ by object, the general methods are similar. For example, the procedure for deleting a federation entity is similar to the procedure for deleting a user directory connection.

The lists within each sub-category let you manipulate an object. You can do one of the following:

- Create a new object.
- Select an existing object from a list of objects and modify it.
- Use the Action button to perform a task on an object.

New Object Creation

If you select a sub category from one of the configuration tabs, an object list is displayed. From an object list, you can create an object.

For example, to establish a new partnership, select the Federation tab to display the View Federation Partnerships window. Click Create Partnership in the Federation Partnership List.

After you click a button to create an object, the appropriate dialog or configuration wizard is displayed to guide you through the procedure.

Objects Lists

When you view a configuration tab or sub-category in the UI, a list of associated objects is also displayed. You can click the link of any object in the list to see detailed information about that object.

For example, from the Federation tab, if you select Entities, the View Federation Entities page displays a list of federation entities.

Action Button for Object Lists

The Action button lies to the left of any object in an object list. Clicking the button displays a menu of tasks you can perform on an object.

The Action button offers different tasks depending on the object.

Filtering Object Lists

If an object type has a particularly long list of configured entries, you can filter which entries are displayed, making the list easier to read. For example, you can search for all configured partnerships whose status is active, and these will appear in the Federation Partnership List.

To specify a search filter

1. Click a main configuration tab.

   The Filter and List group boxes are displayed.

2. Configure the search in the **Filter** *object* group box, using the following guidelines:

   a. Select what you want to search on in the Search For field.

      Examples: For a certificate, you may select Alias as the operand. For a federation entity, you may select Entity Type as the operand.

   b. Choose an operator selected from the pull-down menu in the middle field.

      Examples: =, begins with, contains, ends with; on or before (only option for Expiration Date)

   c. Do one of the following:

      - For operands except Expiration Date, enter a string (do not use quotes) in the last field. This is the value of the search filter.
      - For Expiration Date, select the calendar icon to the right of the field and choose a date. You can enter the date manually, but selecting a date from the calendar ensures the correct date format.

      Notes:

      - To retrieve the full list, leave the third field blank. You may also enter <ANY> or an asterisk (*).
      - You cannot use the asterisk as an embedded wildcard character. For example, you can enter the asterisk on its own, but you cannot enter **partner*** as a value.

3. Click Go to initiate the search.

Page Displays

When you view a list for any object in the UI, only ten records are displayed by default. To the bottom right of the list, choose the greater-than symbol (>) to move to the next page. Choose the double greater-than (>>) symbol to go to the end of the list.

Note: You cannot change the default number of ten records displayed per page.

You can select the Show All link in the top right corner of the list to display all entries in one window.

Wizards for Configuring Objects

There are configuration wizards for many objects in the UI. Wizards are visible when you create or edit an entity or partnership, when you import a certificate, and when you import a metadata file.

The UI wizards help guide you through the steps for configuring a particular object. If you do not fill in the required settings for a given step, you receive a message toward the top of the dialog, which tells you to fill in the missing information. You cannot move to the next step until all required fields are complete.

Log in to the Administrative UI

An administrator configures federation entities through the Administrative UI. You can configure multiple administrators with different privilege levels to access the Administrative UI. Review instructions for configuring multiple administrators.

Follow these steps:

1. Enable Java Script in the browser. JavaScript is required to open the Administrative UI.
2. Follow the instructions for your platform:

   Windows

   Select Start, All Programs, CA, Federation Standalone, CA SiteMinder® Federation Standalone Administrative UI.

   UNIX

   Open a web browser and enter the following URL:
   http://*fed_server:ui_port*/ca/federation/adminui

   fed_server:ui_port

   Specifies the fully qualified domain name of the server where CA SiteMinder® Federation Standalone is installed, including the port for the UI. The default port is 8888.

   Example:

   http://fed1.example.com:8888/ca/federation/adminui

   The login window appears.
3. Enter the user name and password and click Log in.

   Important! The user name for the default administrator is always **admin**. You cannot change it. The default administrator password is set during installation.

   The Administrative UI launches.

UI Login Password Conditions with Active Directory

If you configure Active Directory as the user store for administrative authentication, note the following password conditions when logging in to the Administrative UI.

- When the user is Disabled, the Administrative UI does not allow the user to log in. The system displays a message saying "Error: Invalid user name or password."
- When the user is Expired, the Administrative UI does not allow the user to log in. The system displays a message saying "Error: Invalid user name or password."
- When the user has the attribute "User must change password at next logon" set, the Administrative UI allows the user to log in. The Administrative UI does not handle password management.

Chapter 3: Getting Started with a Simple Partnership

This section contains the following topics:

Basic SAML 2.0 Partnership (see page 37)
Sample Federation Network (see page 38)
Configure the IdP Partner (see page 39)
Configure the SP Partner (see page 47)
Activate the Partnership (see page 54)
Test the Partnership (POST Profile) (see page 55)
Enable Signature Processing (see page 56)
Add Single Logout (see page 60)
Set Up the Artifact Profile for SSO (see page 63)
Configuration Procedures Beyond the Simple Partnership (see page 67)

Basic SAML 2.0 Partnership

One way to get started with CA SiteMinder® Federation Standalone is by configuring a partnership. This chapter describes how to set up a basic SAML 2.0 federation partnership—single sign-on with SAML 2.0 POST profile. By starting with a basic configuration, you can complete the least number of steps to see how the product works.

Note: This partnership focuses on SAML 2.0; however, the overall process is the same for SAML 1.1. The configuration settings at each step of the partnership can differ depending on the SAML protocol.

The chapter also describes the configuration of additional features, such as digital signing and single logout to reflect a real production environment. You can also add the Artifact binding to the configuration.

The sample network used in this chapter presupposes that the product is installed at both sites in the partnership. However, you can have CA SiteMinder® Federation Standalone at one site and a different SAML-compliant product at the other site and still engage in a partnership.

With CA SiteMinder® Federation Standalone at both sites, you have to understand the perspective from which you are configuring a partnership. To configure a complete partnership, you begin by defining a *partnership definition* at each site, one for each direction of communication from a given site. For example, if the local site is the Identity Provider (IdP), you configure the local IdP-to-remote SP partnership. This configuration is one partnership definition. To complete the partnership configuration, you configure the reciprocal local SP-to-remote IdP partnership at the SP, at the local SP.

The partnership definition always distinguishes the local and remote entities. The local entity is the entity at the site from where you are configuring CA SiteMinder® Federation Standalone. It is not necessarily the same system on which CA SiteMinder® Federation Standalone is installed, but the same domain. The remote entity is the entity at a partner that resides in a different domain from where you are configuring CA SiteMinder® Federation Standalone.

The following process shows the steps for creating the basic CA SiteMinder® Federation Standalone partnership when CA SiteMinder® Federation Standalone is at both sites:

1. Establish a user directory connection.
2. Create the local and remote entities.
3. Configure the local IdP-to-SP partnership definition at the IdP site.
4. Configure the local SP-to-IdP partnership definition at the SP site.
5. Activate the partnership.
6. Test the partnership.

Sample Federation Network

The initial partnership you are creating represents the following sample network:

The Business Partners

- Identity Provider named IdP1
- Service Provider named SP1

SAML Profiles and Features

- SAML 2.0 with POST profile
- Single sign-on
- No signature processing
- FIPS_COMPAT mode

CA SiteMinder® Federation Standalone Deployment Mode

Standalone—no SiteMinder Connector

SSO Service URL at the IdP

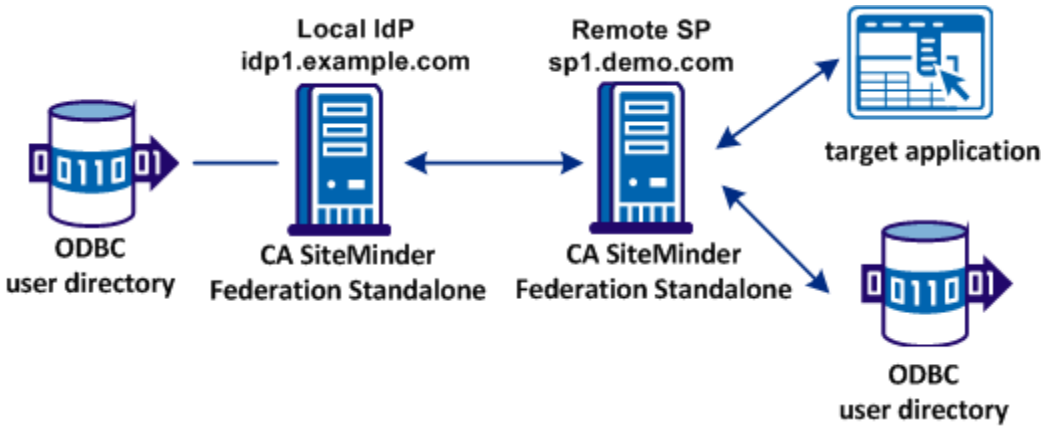
http://idp1.example.com:9090/affwebservices/public/saml2sso

Assertion Consumer Service URL at the SP

http://sp1.demo.com:9091/affwebservices/public/saml2assertionconsumer

Note: You need two systems with CA SiteMinder® Federation Standalone installed to implement this sample network.

The following figure shows the sample partnership.



Configure the IdP Partner

The configuration process that follows is from the perspective of an administrator at IdP1. Therefore, IdP1 is the local IdP.

The following process establishes the IdP partner:

1. Log on to the Administrative UI.
2. Establish a user directory connection.
3. Identify the IdP and SP entities.
4. Click Create Partnership, SAML2 IdP->SP.
5. Follow the Partnership wizard and configure the minimum required settings.

Establish a User Directory Connection

Before you can establish a partnership, define a connection to a user directory.

The procedures that follow illustrate a connection to an ODBC user directory using the default data source that is installed with the product.

Important! The CA FedManager Data Source is where federation system policies are stored. For this example, use this data source as a user directory; however, in a production environment use a different data source.

To use this data source:

- Set up the schema for the sample users for the data source.
- Establish a connection to the directory.

Set up the Sample Users for the Data Source

Set up the sample users for the data store by importing the ODBC schema and sample data.

The product provides script files to create the schema and data for storing sample users in the CA FedManager data source. You can store this data in the same SQL Server or Oracle database that you specified when installing CA SiteMinder® Federation Standalone.

Follow these steps:

1. Navigate to the following directory:

   Windows (default location): *federation_install_dir*\siteminder\db\SQL

   UNIX: *federation_install_dir*/siteminder/db/sql

2. Import the necessary schema files to populate the database with sample users. Use the tool for your database to perform the import.

   Import the following files:

   - smsampleusers_sqlserver.sql

     Creates the schema for sample users in a SQL server database and populates the database with sample users.

   - smsampleusers_oracle.sql

     Creates the schema for sample users in an Oracle database and populates the database with sample users.

   For example, if you look in the script, you can see a sample user named GeorgeC with a password of siteminder.

3. After the schema is imported, connect to the directory.

Connect to the ODBC Directory

After you import the proper schema to populate the ODBC user directory, establish the connection to the user directory.

Follow these steps:

1. Log in to the Administrative UI by opening up a web browser and entering the following URL:

   http://idp1.example.com:8888/ca/federation/adminui

   CA SiteMinder® Federation Standalone is installed with idp1.example.com as the server name. In the browser, map this host name to the IP address where CA SiteMinder® Federation Standalone is installed.

   Note: Verify that JavaScript is enabled in the browser to open the Administrative UI.

2. Select the User Directory tab from the Administrative UI.

   The View User Directories dialog displays.

3. Click Connect to ODBC.

   The Connect to ODBC dialog opens.

4. Complete the following required fields in the Configure ODBC User Directory group section:

   Directory Name

   FedSQL

   Data Source

   CA FedManager Data Source

5. Complete the following fields in the Connection Credentials group section:

   Require Credentials to Connect

   Select check box

   User Name

   Enter the name used to access your database.

   Password

   Enter the password used to access your database.

   Confirm Password

   Enter the database password again.

6. Complete the following field in the Directory Fields group section:

   Universal ID Column

   Enter the name of the ODBC directory attribute used as the Universal ID. This value can be passed to other applications that communicate with CA SiteMinder® Federation Standalone to maintain the identity of the user. This field is required when the CA SiteMinder® Connector is enabled.

7. Click Save.

   You return to the View User Directories dialog.

8. Select Action, Test Connection to help ensure that CA SiteMinder® Federation Standalone can connect to the user directory.

   You receive a message indicating whether the connection is successful.

Continue by configuring the IdP and SP entities.

Configure the Partnership Entities

After establishing the user directory connection, you should identify the local and remote sides of the partnership. In the Administrative UI, each partner is referred to as an entity.

The following procedures tell you what values to provide for the local and remote entities. However, in a real network configuration, it may be common that each side creates a local entity, exports the local entity to a metadata file, then exchanges the files so that each side can define the remote entity.

Follow these steps:

1. From the Federation tab, select Entities.
2. Click Create Entity.
3. Make the following selections in the first step of the entity wizard then click Next.

   Entity Location

   Local

   New Entity Type

   SAML2 IDP

4. Complete the fields in the second step of the wizard as follows then click Next.

   Entity ID

   idp1

   This value identifies the entity to the partner.

   Entity Name

   idp1

   This value identifies the entity object internally in the CA SiteMinder® Federation Standalone database. The partner is not aware of this value.

   Base URL

   http://idp1.example.com:9090

   Leave the other settings as they are.

   Note: The Entity Name can be the same value as the Entity ID, but the value must then not be shared with any other entity at the site.

5. Review the settings in the last step and click Finish.

You return to the View Federation Entities window. Configure the remote partner.

To create the Remote SP Entity

1. Begin at the View Federation Entities window.
2. Click Create Entity in the Federation Entity List.

   The Create Entity dialog displays.

3. Make the following selections in the first step of the entity wizard then click Next.

   Entity Location

   Remote

   New Entity Type

   SAML2 SP

4. Complete the fields in the second step of the wizard as follows then click Next.

   Entity ID

   sp1

   This value identifies the entity to the partner.

   Entity Name

   sp1

   This value identifies the entity object internally in the CA SiteMinder® Federation Standalone database. The partner is not aware of this value.

Assertion Consumer Service URL Group Box

Index

0

Binding

HTTP-Post

URL

http://sp1.demo.com:9091/affwebservices/public/saml2assertionconsumer

Default

Select the checkbox in this column for the entry row.

Leave the other settings as they are.

5. Review the settings in the last step and click Finish.

The remote SP entity is configured.

After the local and remote entity are configured, you can now create a partnership.

Create the IdP-to-SP Partnership

After creating the partnership entities, follow the partnership wizard to configure the IdP ->SP partnership. The first is to provide the name and other basic information for the partnership.

Follow these steps:

1. Select the Federation tab.
2. Click Create Partnership, SAML2 IdP -> SP.

   Selecting this option indicates that you are the local IdP.

   You come to the first step in the partnership wizard.

3. Complete the fields with the following values:

   Partnership Name

   TestPartnership

   Local IDP ID

   idp1

   (selected from the pull-down list)

   Remote SP ID

   sp1

   (selected from the pull-down list)

   Base URL

   http://idp1.example.com:9090

   This value should be provided by default.

   Skew Time (Seconds)

   Accept the default

4. Move the ODBC directory (FedSQL) from the Available Directories box to the Selected Directories box.
5. Click Next to go to the Federation Users step.

Specify Federation Users for Assertion Generation

In the Federation Users dialog, select the users for which the IdP generates assertions.

Follow these steps:

1. Accept the defaults.
2. Click Next to continue.

By accepting the defaults, you indicate that CA SiteMinder® can generate assertions for all users in the user directory.

Add a Name ID to the Assertion

The Assertion Configuration step lets you specify the format and value of the NameID and the attributes that identify a user. These attributes are included in the assertion.

Note: NameID is always included in the assertion.

In this configuration, specify only the Name ID. Do not add any other attributes.

Follow these steps:

1. From the Assertion Configuration step, enter values for the following fields:

   Name ID Format

   Unspecified

   Name ID Type

   Static

   Value

   GeorgeC

2. Click Next to move on and set up single sign-on (SSO).

Set Up Single Sign-on

To establish single sign-on between partners, configure the SSO settings.

Follow these steps:

1. Begin at the SSO and SLO step in the partnership wizard.
2. Accept the default (Basic) for the Local Authentication Type and Authentication Class fields.
3. Select HTTP-POST for the SSO Binding field.
4. Assuming you created the remote SP entity already, the value for the Assertion Consumer URL is filled in.
5. Click Next to move to the Signature and Encryption step.

Disable Signature Processing

For the purposes of this simple partnership, disable signature processing. However, in a production environment, the Identity Provider must sign assertions.

Follow these steps:

1. From the Signature and Encryption step, select Disable Signature Processing.
2. Click Next to move to the next step.

Confirm the IdP-to-SP Partnership Settings

You have completed the partnership definition for one side of the federation partnership. Verify the settings.

Follow these steps:

1. In the Confirm dialog, review the settings for the partnership.
2. To modify a setting, click Modify in any of the sections.
3. Click Finish when you are satisfied with the configuration.

The IdP side of the partnership is complete. Define the SP side of the partnership on a different system than the IdP system.

Configure the SP Partner

The configuration process that follows is from the perspective of an administrator at the SP, in this example, SP1. Therefore, SP1 is the local SP.

The following process establishes the SP partner.

1. Log on to the Administrative UI.
2. Establish a user directory connection.
3. Identify the IdP and SP entities.
4. Click Create Partnership, SAML2 SP->IdP.
5. Follow the Partnership wizard and configure the minimum required settings.

Establish a User Directory Connection

Before you can establish a partnership, define a connection to a user directory.

The procedures that follow illustrate a connection to an ODBC user directory using the default data source that is installed with CA SiteMinder® Federation Standalone.

Important! The CA FedManager Data Source is where CA SiteMinder® Federation Standalone policies are stored. For this example, use this data source as a user directory; however, in a production environment use a different data source.

To use this data source:

- Set up the schema for the sample users for the data source.
- Establish a connection to the directory.

Set up the Sample Users for the Data Source

Set up the sample users for the data store by importing the ODBC schema and sample data.

The product provides script files to create the schema and data for storing sample users in the CA FedManager data source. You can store this data in the same SQL Server or Oracle database that you specified when installing CA SiteMinder® Federation Standalone.

Follow these steps:

1. Navigate to the following directory:

   Windows (default location): *federation_install_dir*\siteminder\db\SQL

   UNIX: *federation_install_dir*/siteminder/db/sql

2. Import the necessary schema files to populate the database with sample users. Use the tool for your database to perform the import.

   Import the following files:

   - smsampleusers_sqlserver.sql

     Creates the schema for sample users in a SQL server database and populates the database with sample users.

   - smsampleusers_oracle.sql

     Creates the schema for sample users in an Oracle database and populates the database with sample users.

   For example, if you look in the script, you can see a sample user named GeorgeC with a password of siteminder.

3. After the schema is imported, connect to the directory.

Connect to the ODBC Directory

After you import the proper schema to populate the ODBC user directory, establish the connection to the user directory.

Follow these steps:

1. Log in to the Administrative UI by opening up a web browser and entering the following URL:

   http://idp1.example.com:8888/ca/federation/adminui

   CA SiteMinder® Federation Standalone is installed with idp1.example.com as the server name. In the browser, map this host name to the IP address where CA SiteMinder® Federation Standalone is installed.

   Note: Verify that JavaScript is enabled in the browser to open the Administrative UI.

2. Select the User Directory tab from the Administrative UI.

   The View User Directories dialog displays.

3. Click Connect to ODBC.

   The Connect to ODBC dialog opens.

4. Complete the following required fields in the Configure ODBC User Directory group section:

   Directory Name

   FedSQL

   Data Source

   CA FedManager Data Source

5. Complete the following fields in the Connection Credentials group section:

   Require Credentials to Connect

   Select check box

   User Name

   Enter the name used to access your database.

   Password

   Enter the password used to access your database.

   Confirm Password

   Enter the database password again.

6. Complete the following field in the Directory Fields group section:

   Universal ID Column

   Enter the name of the ODBC directory attribute used as the Universal ID. This value can be passed to other applications that communicate with CA SiteMinder® Federation Standalone to maintain the identity of the user. This field is required when the CA SiteMinder® Connector is enabled.

7. Click Save.

   You return to the View User Directories dialog.

8. Select Action, Test Connection to help ensure that CA SiteMinder® Federation Standalone can connect to the user directory.

   You receive a message indicating whether the connection is successful.

Continue by configuring the IdP and SP entities.

Identify the Partnership Entities

After establishing the user directory connection, you should identify the local and remote sides of the partnership. In the Administrative UI, each partner is referred to as an entity.

The following procedures tell you what values to provide for the local and remote entities. However, in a real network configuration it may be common that each side creates a local entity, exports the local entity to a metadata file, then exchanges the files so that each side can define the remote entity.

Follow these steps:

1. From the Federation tab, select Entities.
2. Click Create Entity.

   The Create Entity dialog displays.

3. Make the following selections in the first step of the entity wizard then click Next.

   Entity Location

   Local

   New Entity Type

   SAML2 SP

4. Complete the fields in the second step as follows then click Next.

   Entity ID

   sp1

   This value identifies the entity to the partner.

   Entity Name

   sp1

   This value identifies the entity object internally in the CA SiteMinder® Federation Standalone database. The partner is not aware of this value.

   Base URL

   http://sp1.demo.com:9091

   Note: The entity ID and name must be the same as you specified for the remote SP entity at the Identity Provider.

5. Review the settings and click Finish.

You return to the View Federation Entities window. Configure the remote partner.

To create the remote IdP

1. Begin at the View Federation Partnerships window.
2. Click Create Entity in the Federation Entity List.

   The Create Entity dialog displays.

3. Make the following selections in the first step of the entity wizard then click Next.

   Entity Location

   Remote

   New Entity Type

   SAML2 IDP

4. Complete the fields in the second step of the wizard as follows:

   Entity ID

   idp1

   This value identifies the entity to the partner.

   Entity Name

   idp1

   This value identifies the entity object internally in the CA SiteMinder® Federation Standalone database. The partner is not aware of this value.

   Note: The entity ID and name must be the same as on the Identity Provider side.

SSO Service URL Group Box

Binding

HTTP-Redirect

URL

http://idp1.example.com:9090/affwebservices/public/saml2sso

5. Review the settings and click Finish.

After the local entity and remote entity are configured, you can create a partnership.

Create the SP-to-IdP Partnership

After creating the partnership entities, follow the Partnership wizard to configure the necessary components of the SP -> IdP partnership.

Follow these steps:

1. Select the Federation tab.
2. Click Create Partnership, SAML2 SP->IdP.

   You come to the first step in the Partnership wizard.
3. Complete the fields with the following values:

   Partnership Name

   DemoPartnership

   Local SP ID

   sp1

   Remote IDP ID

   idp1

   Skew Time (Seconds)

   Accept the default
4. Move the ODBC directory (FedSQL) from the Available Directories box to the Selected Directories box.
5. Click Next to go to the User Identification step.

Specify the User Identification Attribute

Designate which attribute from the assertion should be used to identify a user. This identity attribute value is used in the user disambiguation process, that is, the process of locating the user record in the SP's user directory.

Follow these steps:

1. Go to the User Identification step.
2. Accept the default, Use Name ID, in the Choose Identity Attribute from Assertion group box.
3. In the Map Identity Attribute to User Directories group box, enter the following:

   ODBC Search Specification

   Name=%s

   This entry instructs CA SiteMinder® Federation Standalone to replace the variable (%s) with the value of the Name ID attribute from the assertion and match it with the Name column in the sample users database. If a match is found, the user is disambiguated and allowed to access the target resource.
4. Click Next to configure single sign-on.

Configure Single Sign-on

To establish single sign-on between partners, configure the SSO settings.

Follow these steps:

1. Begin at the SSO and SLO step.
2. Select HTTP-POST for the SSO Binding field.
3. Specify the target resource at the SP in the Target field.

   In this sample partnership, this target is http://spapp.demo.com:80/spsample/welcome.html
4. Select No Data for the Redirect Mode field.
5. Assuming you have created the remote IdP, the value for the SSO Service URL is filled in.
6. Click Next to move to the Signature and Encryption step.

Disable Signature Processing

For the purposes of this simple partnership, disable signature processing. However, in a production environment, the Identity Provider must sign assertions.

Follow these steps:

1. From the Signature and Encryption step, select Disable Signature Processing.
2. Click Next to move to the next step.

Confirm the SP Partner Settings

You have completed the partnership for the local SP side of the federation partnership.

Follow these steps:

1. In the Confirm dialog, review the settings for the SP partner.
2. To modify a setting, click Modify in the appropriate section.
3. Click Finish when you are satisfied with the configuration.

The SP side of the partnership is now configured.

Activate the Partnership

With each side of the partnership defined, you can now activate the partnership.

CA SiteMinder® Federation Standalone is installed at both sites in the partnership so you must activate the partnership at the IdP and SP.

Follow these steps:

1. From the Federation tab, select Partnerships

   The View Federation Partnerships window displays.
2. Find the entry in the Federation Partnership List that you want to activate. Ensure the value in the Status column is DEFINED If the status is INCOMPLETE, you have to edit the partnership and ensure that all the required settings are configured.
3. Select Action, Activate next to the partnership entry that you want to activate.

   The Confirm Activate dialog displays.
4. Click Yes in the Confirm Activate dialog.

   The partnership is activated and the value in the Status column should now be ACTIVE.

Test the Partnership (POST Profile)

After the partnership is configured, test single sign-on between the two partners.

Testing involves:

- Creating a web page to initiate single sign-on.
- Creating a target web page that serves as the requested federated resource.
- Testing single sign-on.

After you test the basic partnership, you can make more changes to the sample configuration.

Create a Web Page to Initiate Single Sign-on

For testing purposes, create your own html page with a link that initiates single sign-on. You can initiate single sign-on from the IdP or SP. This example illustrates SP-initiated single sign-on.

Follow these steps:

1. Create the sample HTML page at the SP site. Include a hard-coded link to the AuthnRequest service at the SP, as follows:

   <a href="http://sp1.demo.com:9091/affwebservices/public/
   saml2authnrequest?ProviderID=idp1.example.com>
   Link to Test POST Single Sign-on

   This link instructs the AuthnRequest Service to redirect the user to the specified Identity Provider to retrieve the authentication context.

2. Save the web page under the name testsso.html.
3. Copy testsso.html to the web server document root directory, under a subfolder named /spsample.

   For this sample network, the target web server is http://spapp.demo:80.

Create a Target Resource

The last step that is required to test single sign-on is to create a target resource.

Follow these steps:

1. Create the sample HTML page at the SP site and include a message, such as:

   Welcome to SP1

   Single Sign-on is successful

2. Save the web page under the name welcome.html.
3. Copy welcome.html to the web server document root directory, under the subfolder /spsample.

   For this sample network, the target web server is http://spapp.demo.com:80.

Test POST Single Sign-on

After you have set up the sample web pages, test single sign-on and verify that that partnership configuration is successful.

Follow these steps:

1. Be sure that both sides of the partnership are activated in the Administrative UI.
2. Open up a browser.
3. Enter the URL for the web page that includes the link to trigger single sign-on. For this example, enter the following url:

   http://spapp.demo.com:80/spsample/testsso.html

   Note: In this sample network, CA SiteMinder® Federation Standalone is deployed in standalone mode, therefore, the target web server is a different server than the one where CA SiteMinder® Federation Standalone resides.

   Upon entering the URL, a page appears with a link that reads Link to Test POST Single Sign-on.
4. Click **Link to Test POST Single Sign-on**.

   Single sign-on is initiated. The user is redirected from the AuthnRequest Service at the SP to the Single Sign-on Service at the Identity Provider.

After the Identity Provider authenticates the user and establishes a session, it directs the user back to the target resource at the Service Provider, which is welcome.html. The sample welcome page that you created at the SP, lets you know the single sign-on was successful.

Enable Signature Processing

Digitally signing assertions is required in a SAML 2.0 POST single sign-on. For signing and verification tasks, a private key/certificate pair is used.

Before any transaction or runtime actions, an administrator at IdP1 sends a file with certificate data to SP1. This file contains a certificate (public key) associated with the private key that the IdP1 uses to sign assertions. An administrator at SP1 adds the certificate to its certificate data store.

When the single sign-on transaction occurs, IdP1 signs the assertion with its private key. SP1 receives the assertion and verifies the assertion signature using the certificate in the certificate data store.

The following procedures explain how to set up signing at each site.

Configure Signature Processing at the IdP

For POST single sign-on, Idp1 is required to sign assertions. It uses the private key in the certificate data store to sign assertions.

Note: The example assumes that you have a file from which you to import keys and certificates, or that you already have private keys and certificates for signing and verification tasks.

Follow these steps:

1. From the UI, click the Federation tab and select Partnerships.

   The View Federation Partnerships window displays.

2. Select Action, Deactivate next to the entry for TestPartnership, which is the IdP ->SP partnership.

   Deactivate a partnership before editing it.

3. Click Action, Modify next to the entry for TestPartnership.

   The dialog for the first step of the Partnership wizard opens.

4. Click the Signature and Encryption step in the partnership wizard.

5. In the Signature group box:

   a. Deselect Disable Signature Processing.

   b. Click Import next to the Signing Private Key Alias field.

      The Import Certificate/Private Key window opens.

6. Complete the import wizard as follows:

   a. Select the file from where you are importing the private key/certificate pair.

   b. If the file is a pkcs#12 file, supply the password to encrypt the file.

   c. Select the certificate entry from the file that you want to import and enter a value for the Alias, such as cert1.

   d. Confirm the selection and click Finish.

   You return to the View Federation Partnerships window.

7. Select Action, Modify for the partnership entry.

8. Go to the Signature and Encryption step. In the dialog, the key/certificate that you imported is now available from the Signing Private Key Alias drop-down list.
9. Select the alias for cert1 and click Next.
10. Review the settings in the Confirm dialog and click Finish.

    You return to the View Federation Partnerships window.
11. Reactivate the partnership by selecting Action, Activate next to the TestPartnership entry in the Federation Partnership List.
12. Restart the federation services according to your operating environment.

    - **Windows**

      Use the stop and start shortcuts as follows. If you logged in as a network user and not a local administrator, right-click the shortcut and select Run as administrator.

      a. Start, All Programs, CA, Federation Standalone, Stop services

      b. Start, All Programs, CA, Federation Standalone, Start services

    - **UNIX**

      a. Open a command window.

      b. Run the following scripts:

      *federation_install_dir/*fedmanager.sh stop

      federation_install_dir/fedmanager.sh start

      Note: Do not stop and start the services as the root user.

    Restarting the federation services makes the system aware of the changes to signing.

Signature processing is now configured at the IdP.

Configure Signature Processing at the SP

SP1 is required to verify the signature of an assertion. Before a transaction, SP1 has to have the certificate (public key) from IdP1. This is the certificate that is associated with the private key that IdP1 uses to sign the assertion.

This certificate must be imported into SP1 certificate data store.

Follow these steps:

1. From the Administrative UI, click the Federation tab and select Partnerships.

   The View Federation Partnerships window displays.

2. Select Action, Deactivate next to the entry for DemoPartnership.

   Deactivate a partnership before editing it.

3. Click Action, Modify next to the entry for DemoPartnership.

   The dialog for the first step of the Partnership wizard opens.

4. Click the Signature and Encryption step in the Partnership wizard.

5. In the Signature group box:

   a. Deselect Disable Signature Processing.

   b. Click Import next to the Verification Certificate Alias field.

      The Import Certificate/Private Key window opens.

6. Complete the import wizard as follows:

   a. Select the file from where you are importing the certificate.

   b. Select the certificate entry from the file that you want to import and enter a value for the Alias, such as cert1.

   c. Confirm the selection and click Finish.

   You return to the View Federation Partnerships window.

7. Select Action, Modify for the partnership entry.

8. Go to the Signature and Encryption step. In the dialog, the key/certificate that you imported is now available from the Signing Private Key Alias drop-down list.

9. Select the alias, cert1 for the certificate and click Next.

10. Review the settings in the Confirm dialog and click Finish.

    You return to the View Federation Partnerships window.

11. Reactivate the partnership by selecting Action, Activate next to the DemoPartnership entry in the Federation Partnership List.
12. Restart the federation services according to your operating environment.
    - **Windows**

      Use the stop and start shortcuts as follows. If you logged in as a network user and not a local administrator, right-click the shortcut and select Run as administrator.

      a. Start, All Programs, CA, Federation Standalone, Stop services

      b. Start, All Programs, CA, Federation Standalone, Start services

    - **UNIX**

      a. Open a command window.

      b. Run the following scripts:

      *federation_install_dir/*fedmanager.sh stop

      federation_install_dir/fedmanager.sh start

      Note: Do not stop and start the services as the root user.

    Restarting federation services makes the system aware of the changes to signing.

Signature verification is now configured at the SP.

Add Single Logout

The single logout protocol (SLO) results in the simultaneous end of all user sessions for the browser that initiated the logout. Configuring single logout helps ensure that no sessions are left open for unauthorized users to gain access to resources at the Service Provider.

Configure Single Logout at the IdP

Configure single logout at Idp1.

Follow these steps:

1. Log in to the Administrative UI.
2. Click Federation, Partnerships.

   The View Federation Partnerships window displays.
3. Select Action, Deactivate next to the entry for TestPartnership.

   Deactivation is required before editing.
4. Click Action, Modify next to the entry for TestPartnership.

   The dialog for the first step of the partnership opens.
5. Click the SSO and SLO step.
6. In the SLO section, select the HTTP-redirect for the SLO Bindings to enable single logout.
7. Click Add Row in the SLO Service URLs table and complete the following:

   SLO Location URL

   http://sp1.demo.com:9091/affwebservices/public/saml2slo

   This link indicates that the single logout request is sent to the remote SP.

   SLO Confirm URL

   http://idp1.example.com:9090/idpsample/SLOConfirm.html

   This link is the confirmation page at the site that initiated single logout, in this case, IdP1. The user is redirected to this page when single logout completes successfully.
8. Select the row you configured by clicking the option button in the Select column.
9. Click the Confirm step in the wizard and review the configuration.
10. Click Finish.

    You return to the View Federation Partnerships window.
11. Reactivate the partnership by selecting Action, Activate next to the TestPartnership entry in the Federation Partnership List.

Single logout is now added to the configuration at IdP1.

Configure Single Logout at the SP

Configure single logout at SP1.

Follow these steps:

1. Log in to the Administrative UI.
2. Click Federation, Partnerships.
3. Select Action, Deactivate next to the entry for Demo Partnership.

   You must deactivate a partnership prior to editing it.
4. Click Action, Modify next to the entry for DemoPartnership.

   The dialog for the first step of the partnership wizard opens.
5. Click the SSO and SLO step.
6. In the SLO group box, select the HTTP-redirect for the SLO Bindings to enable single logout.
7. Click Add Row in the SLO Service URLs table, if there is no row available complete the following:

   SLO Location URL

   http://idp1.example.com:9090/affwebservices/public/saml2slo

   This is the link where the single logout request will be sent.

   SLO Confirm URL

   http://sp1.demo.com:9091/spsample/SLOConfirm.html

   This is the single logout confirmation page at the site that initiated the logout.
8. Select the row you just configured by clicking the radio button in the Select column.
9. Click the Confirm step in the wizard and review the configuration.
10. Click Finish.

    You return to the View Federation Partnerships window.
11. Reactivate the partnership by selecting Action, Activate next to the DemoPartnership entry in the Federation Partnership List.

Single logout is now configured at the SP.

Test Single Logout

After you configure single logout, test it. For this test, single logout is initiated at SP1.

Initiating single logout from the SP requires that you have two web pages to initiate and confirm single logout.

- Using welcome.html, add a link to this page that directs the browser to the Single Logout Service at IdP1. This link has the following syntax:

  Log Me Out

- Create a confirmation page named SLOConfirm.html with a logout confirmation message, such as:

  You have successfully logged out

Copy both these pages to your web server root directory under the subfolder /spsample.

Note: Complete an SSO transaction so you can test SLO.

Follow these steps:

1. Verify that both sides of the partnership are activated in the Administrative UI.
2. Configure and test single sign-on according to the previously documented instructions.

   If single sign-on is successful, the welcome page is displayed in the browser.
3. Keep the browser open and click the link **Log Me Out** on the welcome page.

   If successful, you are redirected to the confirmation page that displays the message:

   You have successfully logged out.

Set Up the Artifact Profile for SSO

The basic partnership began with HTTP-POST binding for single sign-on. However, your partnership can use the SAML 2.0 Artifact profile.

The configuration for the HTTP-Artifact binding is the same as the configuration for POST binding, until the SSO and SLO steps in the wizard.

Configure Artifact SSO at the IdP

This procedure shows you how to configure HTTP-Artifact profile for SSO.

Follow these steps:

1. From the Administrative UI, click the Federation tab and select Partnerships.
2. Select Action, Deactivate next to the entry for TestPartnership.

   Deactivation is required before editing.
3. Click Action, Modify next to the entry for TestPartnership.

   The dialog for the first step of the partnership wizard opens.
4. Click the SSO and SLO step.
5. Keep the existing settings in the Authentication group box.
6. In the SSO group box, do the following:
   a. Check HTTP-Artifact for the SSO Binding field.
   b. Change the binding in the Assertion Consumer Service URLs table to HTTP-Artifact. The URL can remain the same as was used for POST profile.
7. In the Back Channel group box, select the following:

   Authentication Method

   No Auth
8. Skip the SLO and IDP Discovery group boxes.
9. Click the Confirm step and review the configuration.
10. Click Finish to complete the configuration.

Artifact binding is now configured at Idp1.

Configure Artifact SSO at the SP

This procedure describes how to configure the HTTP-Artifact profile for SSO.

Follow these steps:

1. From the Administrative UI, click the Federation tab and select Partnerships.

   The View Federation Partnerships window displays.
2. Select Action, Deactivate next to the entry for Demo Partnership.

   Deactivate a partnership before you edit it.
3. Click Action, Modify next to the entry for DemoPartnership.

   The dialog for the first step of the Partnership wizard opens.

4. Click the SSO and SLO step.
5. In the SSO group box, do the following tasks:
   a. Check HTTP-Artifact for the SSO Binding field.
   b. Select No Data for the Redirect Mode field. The URL can remain the same as was used for POST profile.
   c. Do not change the settings for the SSO Service URL.
6. In the SOAP Artifact Resolution URLs group box, click Add Row and enter the following URL to indicate that no authentication is required for the back channel:

   http://idp1.example.com:9090/affwebservices/saml2artifactresolutionnoauth

   Be sure to select this entry by clicking the radio button in the Select column of the table.
7. In the Back Channel group box, select the following option:

   Authentication Method

   No Auth
8. Skip the SLO and Status Redirect URL group boxes.
9. Click the Confirm step and review the configuration.
10. Click Finish to complete the configuration.

Artifact binding is configured at SP1.

Test the Partnership (Artifact SSO)

When each side of the partnership is operating, test single sign-on between the two partners.

When IdP1 receives the request, it generates the artifact. The artifact is then sent to the SP1.

After SP1 receives the artifact, it redirects the request back to IdP1. The IdP retrieves the assertion and returns it to SP1.

Create a Web page to Initiate Single Sign-on (Artifact)

For testing purposes, create your own html page with a link that initiates single sign-on. You can initiate single sign-on from the IdP or SP. This example illustrates SP-initiated single sign-on.

Follow these steps:

1. Create the sample HTML page at the SP site and include a hard-coded link to the AuthnRequest service at the SP, as follows:

   Link for ARTIFACT Single Sign-on

   This link instructs the AuthnRequest Service to redirect the user to the specified Identity Provider to retrieve the user authentication context.

2. Save the web page under the name testartifact.html.
3. Copy testartifact.html to the web server document root directory, under the subfolder /spsample.

   For this sample network, the target web server is http://spapp.demo:80.

Create a Target Resource

The last step that is required to test single sign-on is to create a target resource.

Follow these steps:

1. Create the sample HTML page at the SP site and include a message, such as:

   Welcome to SP1

   Single Sign-on is successful

2. Save the web page under the name welcome.html.
3. Copy welcome.html to the web server document root directory, under the subfolder /spsample.

   For this sample network, the target web server is http://spapp.demo.com:80.

Test Artifact Single Sign-on

After you have set up the sample web pages, test single sign-on and verify that the partnership configuration is successful.

Follow these steps:

1. Verify that both sides of the partnership are activated.
2. Open up a browser.

3. Enter the URL for the web page that triggers single sign-on, as follows:

   http://spapp.demo.com:80/spsample/testartifact.html

   Note: In this sample network, CA SiteMinder® Federation Standalone is deployed in standalone mode, therefore, the target web server is a different server than the one where CA SiteMinder® Federation Standalone resides.

   When entering the URL, a page is displayed with a link that reads Link to Test ARTIFACT Single Sign-on.

4. Click **Link to Test ARTIFACT Single Sign-on** and single sign-on is initiated.

   The user is redirected from the AuthnRequest Service at the SP to the Single Sign-on Service at the Identity Provider.

After the Identity Provider establishes a session, it directs the user back to the target resource at the Service Provider, which is welcome.html. You see the sample welcome page you created at the SP, letting you know that single sign-on was successful.

Configuration Procedures Beyond the Simple Partnership

The simple partnership described in this chapter gives you an overview of configuring federated partnerships using CA SiteMinder® Federation Standalone.

The remaining chapters in the guide provide detailed procedures for every task you can perform with CA SiteMinder® Federation Standalone. For detailed configuration instructions, use these procedures as well as the Help in the Administrative UI.

More information:

Federation Entity Configuration (see page 107)
Partnership Creation and Activation (see page 143)
User Directory Connections for Authentication (see page 73)

Chapter 4: Storing User Session, Assertion, and Expiry Data

This section contains the following topics:

Federation Data Stored in the Session Store (see page 69)
Enable the Session Store (see page 70)
Environments that Require a Shared Session Store (see page 71)

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.

  Note: 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 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.

- 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 Policy Server 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 Identity Provider and Resource Partner.

- Authentication Session Variables Persistence (all profiles)

  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. This mode instructs the Policy Server to store assertion attributes in the session store so they can be supplied as HTTP header variables.

Enable the Session Store

Enable the session store to store data when using SAML artifact for single sign-on, single logout, and enabling the single use of a policy.

Enable the session store from the Policy Server Management Console.

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

Follow these steps: to enable a session store

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: The Use Policy Store database option is disabled. For performance reasons, the session server cannot be run on the same database as the policy store.

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 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.

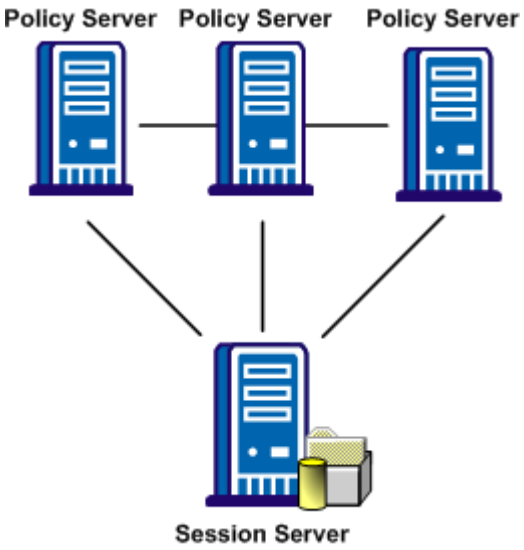
- For SAML 2.0 single logout and WS-Federation sign-out, share the session store at the asserting and relying party across all Policy Servers in the cluster.

  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 5: User Directory Connections for Authentication

This section contains the following topics:

User Directory Management Overview (see page 73)
LDAP Directory Connection (see page 74)
How to Connect to an LDAP User Directory Over SSL (see page 76)
ODBC Directory Connection (see page 85)
Test a User Directory Connection from the Directory List (see page 89)
Create a Common View of the Same User Information Across Directories (see page 89)

User Directory Management Overview

Directory connections resolve how CA SiteMinder® Federation Standalone establishes a context for user identities. The system uses these connections to verify user identities and retrieve user attributes contained in user stores.

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 user's assertion is presented during authentication, the relying party looks in the user directory for the user record.

You configure connections to existing user directories through the User Directory tab in the Administrative UI. You are only establishing a connection to a user directory. You are not configuring a new user directory.

You can configure connections to more than one directory, and the directories do not have to be the same type (LDAP or ODBC).

Important! If you are using the SiteMinder Connector, the user directory must be configured to connect to the same directory that SiteMinder is pointing to, and it must be configured using the same name that SiteMinder uses for that directory.

LDAP Directory Connection

You can establish a connection to an LDAP directory so CA SiteMinder® Federation Standalone can use it as a user store for authentication.

Follow these steps:

1. Click the User Directory tab.
2. Click Connect to LDAP in the User Directory List section.
3. Configure the settings in each section. Parameters marked by red dots are required.

   Note: Click Help for a description of fields, controls, and their respective requirements.
4. Click Failover/Load Balancing if you want to set up either of these features.
5. Click Test Connection to verify that the directory connection is valid.

   You can click View Contents to list the contents of the user directory.

   Note: The View Contents button is displayed only if the Search Root, Start User DN Lookup, End User DN Lookup, and Universal ID Attribute values are set
6. Click Save.

   If your settings are valid, you are redirected to the View User Directories dialog.

   The connection to the LDAP directory is configured.

Load Balancing and Failover for LDAP User Directories

CA SiteMinder® Federation Standalone can distribute LDAP user directory requests over multiple LDAP servers for failover and load balancing.

For load balancing, the system evenly spreads requests over the specified LDAP servers. Coupled with failover, load balancing provides faster, more efficient access to LDAP user directory information.

For failover, the system uses one LDAP server to fulfill requests until that server fails to respond. When the default server does not respond, the system routes the request to the next server configured for failover. This process can be repeated over multiple servers. After the default server is able to fulfill requests again, requests go back to the original server.

Follow these steps:

1. Select the User Directory tab in the UI.
2. Do one of the following:
   - Select Connect to LDAP to create an LDAP directory connection.
   - Select Action, Modify next to an existing LDAP entry you want to edit.

   The User Directory dialog opens.
3. Click Configure Load-balancing or Failover or both in the Configure LDAP User Directory section of the dialog.

   The LDAP Server Load-balancing and Failover table displays.
4. Enter the IP address and port number of in the form, *ip_address*:*port,* in the first Failover Node field. Add the addresses of subsequent directory servers in the remaining fields for failover.

   Note: If you are adding a server for failover, the failover directory must use the same type of communication (SSL or non-SSL) as the primary directory. Both directories share the same port number.

   If you only have one entry in the table, then only failover is supported.
5. To configure another group for load balancing, click Add Row and complete the fields as you did in the previous step.

   You can add the same server multiple times for load balancing, which forces a single system to handle more requests. For example, consider two servers in a group: Server1 and Server2. Server1 is a high-performance server and Server2 is a lesser system. You can add Server1 to the load balancing list twice so that it processes two requests for each request processed by Server2.

Example: Load Balancing and Failover

In this example, a SiteMinder environment contains two user directories, A and B, which must meet the following requirements:

- User directory A must failover to user directory B and load balance with B.
- User directory B must failover to user directory A; and load balance with user directory A.

The configuration requires two load balancing groups.

1. Specify the address for user directory B for the first load balancing group and first failover node.
2. Add a load balancing group by clicking Add row.
3. List user directory B as the first server in the new load balancing group.
4. List user directory A as the second sever in the load balancing group.

The result is two load balancing groups with one server each for failover "A B" and "B A", which load balance each other. If both directories are available, load balancing occurs between the first directories in each group: A and B. If user directory A becomes unavailable, failover occurs to user directory B. This results in user directory B handling all the requests until user directory A becomes available.

How to Connect to an LDAP User Directory Over SSL

Connecting to an LDAP user directory over SSL requires that you configure the system to use the certificate database files.

Follow the instructions in the subsequent sections to configure the connection over SSL.

Note: CA Directory does not support this method of configuring SSL.

Before You Configure an LDAP Connection over SSL

Review the following points before configuring an LDAP user directory connection over SSL:

- Verify that your directory server is SSL-enabled.

  Note: For more information about configuring your directory server to communicate over SSL, refer to the vendor-specific documentation.

- The Policy Server uses a Mozilla LDAP SDK to communicate with LDAP directories. As a result, the database files must be in the Netscape database version file format (cert8.db).

  Important! Do not use Microsoft Internet Explorer to install certificates into your cert8.db database file.

- (Active Directory) Considering the following points:
  - If the user directory connection is configured with the AD namespace, the SSL process that is documented in the subsequent topics does not apply. The AD namespace uses the native Windows certificate repository when establishing an SSL connection. When configuring the AD namespace to communicate over SSL:
    - Verify that the user directory connection is configured for a secure connection. For more information, refer to Configure the User Directory Connection for SSL.
    - On the computer hosting the Active Directory instance, verify that the root CA certificate and the server certificate are added to the service certificate store.

    Note: For more information about configuring Active Directory to communicate over SSL, refer to the Microsoft documentation.

If the user directory connection was configured with the LDAP namespace, complete the process to configure the connection over SSL.

Create the Certificate Database Files

Thecertificate database files must be in the Netscape database file format (cert8.db). Use the Mozilla Network Security Services (NSS) certutil application that is installed with the Policy Server to create the certificate database files.

Note: The following procedure details the specific options and arguments to complete the task. For a complete list of the NSS utility options and arguments, refer to the Mozilla documentation on the NSS project page.

Important! Before running a CA SiteMinder® utility or executable on Windows Server 2008, open the command line window with administrator permissions. Open the command line window this way, even if your account has administrator privileges.

Follow these steps:

1. From a command prompt, navigate to the installation bin directory.

   Example: C:\Program Files\CA\SiteMinder\bin

   Note: Windows has a native certutil utility. Verify that you are working from the Policy Server bin directory, or you can inadvertently run the Windows certutil utility.

2. Enter the following command:

   `certutil -N -d certificate_database_directory`

   -N

   Creates the cert8.db, key3.db, and secmod.db certificate database files.

-d *certificate_database_directory*

Specifies the directory in which the certutil tool is to create the certificate database files.

Note: If the file path contains spaces, bracket the path in quotes.

The utility prompts for a password to encrypt the database key.

3. Enter and confirm the password.

   NSS creates the required certificate database files:

   - cert8.db
   - key3.db
   - secmod.db

Example: Create the Certificate Database Files

```
certutil -N -d C:\certdatabase
```

Add the Root Certificate Authority to the Certificate Database

Add the root Certificate Authority (CA) to make it available for communication over SSL. Use the Mozilla Network Security Services (NSS) certutil application installed with the Policy Server to add the root CA.

Note: The following procedure details the specific options and arguments to complete the task. For a complete list of the NSS utility options and arguments, refer to the Mozilla documentation on the NSS project page.

Important! Before running a CA SiteMinder® utility or executable on Windows Server 2008, open the command line window with administrator permissions. Open the command line window this way, even if your account has administrator privileges.

Follow these steps:

1. From a command prompt, navigate to the Policy Server installation bin directory.

   Example: C:\Program Files\CA\SiteMinder\bin

   Note: Windows has a native certutil utility. Verify that you are working from the bin directory of the NSS utility, or you can inadvertently run the Windows certutil utility.

2. Run the following command to add the root CA to the database file:

   `certutil -A -n alias -t trust_arguments -i root_CA_path -d certificate_database_directory`

   -A

   Adds a certificate to the certificate database.

   -n *alias*

   Specifies an alias for the certificate.

   Note: If the alias contains spaces, bracket the alias with quotes.

   -t *trust_arguments*

   Specify the trust attributes to apply to the certificate when adding it to the certificate database. There are three available trust categories for each certificate, which are expressed in this order: "SSL, email, object signing". Specify the appropriate trust arguments so that the root CA is trusted to issue SSL certificates. In each category position, you may use zero or more of the following attribute arguments.

   p

   Valid peer.

   P

   Trusted peer. This argument implies p.

   c

   Valid CA.

   T

   Trusted CA to issue client certificates. This argument implies c.

   C

   Trusted CA to issue server certificates (SSL only). This argument implies c.

   Important! This is a required argument for the SSL trust category.

   u

   Certificate can be used for authentication or signing.

   -i *root_CA_path*

   Specifies the path to the root CA file. Consider the following:

   - The path must include the certificate name.
   - Valid extensions for a certificate include .cert, .cer, and .pem.

   Note: If the file path contains spaces, bracket the path in quotes.

-d *certificate_database_directory*

Specifies the path to the directory that contains the certificate database.

Note: If the file path contains spaces, bracket the path in quotes.

NSS adds the root CA to the certificate database.

Example: Adding a Root CA to the Certificate Database

```
certutil -A -n "My Root CA"  -t "C,," -i C:\certificates\cacert.cer -d C:\certdatabase
```

Add the Server Certificate to the Certificate Database

Add the server certificate to the certificate database to make it available for communication over SSL. Use the Mozilla Network Security Services (NSS) certutil application installed with the Policy Server to add the server certificate.

Note: The following procedure details the specific options and arguments to complete the task. For a complete list of the NSS utility options and arguments, refer to the Mozilla documentation on the NSS project page.

Important! Before running a CA SiteMinder® utility or executable on Windows Server 2008, open the command line window with administrator permissions. Open the command line window this way, even if your account has administrator privileges.

To add the server certificate to the certificate database

1. From a command prompt, navigate to the Policy Server installation bin directory.

   Example: C:\Program Files\CA\SiteMinder\bin

   Note: Windows has a native certutil utility. Verify that you are working from the bin directory of the NSS utility, or you can inadvertently run the Windows certutil utility.

2. Run the following command to add the root certificate to the database file:

   ```
   certutil -A -n alias -t trust_arguments -i server_certificate_path -d
   certificate_database_directory
   ```

   -A

   Adds a certificate to the certificate database.

   -n *alias*

   Specifies an alias for the certificate.

   Note: If the alias contains spaces, bracket the alias with quotes.

-t *trust_arguments*

Specify the trust attributes to apply to the certificate when adding it to the certificate database. There are three available trust categories for each certificate, which are expressed in this order: "SSL, email, object signing". Specify the appropriate trust arguments so that the certificate is trusted. In each category position, you may use zero or more of the following attribute arguments:

p

Valid peer.

P

Trusted peer. This argument implies p.

Important! This is a required argument for the SSL trust category.

-i *server_certificate_path*

Specifies the path to the server certificate. Consider the following:

- The path must include the certificate name.
- Valid extensions for a certificate include .cert, .cer, and .pem.

Note: If the file path contains spaces, bracket the path in quotes.

-d *certificate_database_directory*

Specifies the path to the directory that contains the certificate database.

Note: If the file path contains spaces, bracket the path in quotes.

NSS adds the server certificate to the certificate database.

Example: Adding a Server Certificate to the Certificate Database

```
certutil -A -n "My Server Certificate" -t "P,," -i C:\certificates\servercert.cer -d
C:\certdatabase
```

Verify that the Certificates are in the Database

List the certificates to verify that they were added to the certificate database. Use the Mozilla Network Security Services (NSS) certutil application that is installed with the Policy Server to create the certificate database files.

Note: The following procedure details the specific options and arguments to complete the task. For a complete list of the NSS utility options and arguments, refer to the Mozilla documentation on the NSS project page.

Important! Before running a CA SiteMinder® utility or executable on Windows Server 2008, open the command line window with administrator permissions. Open the command line window this way, even if your account has administrator privileges.

Follow these steps:

1. From a command prompt, navigate to the Policy Server installation bin directory.

   Example: C:\Program Files\CA\SiteMinder\bin

   Note: Windows has a native certutil utility. Verify that you are working from the bin directory of the NSS utility, or you can inadvertently run the Windows certutil utility.

2. Run the following command:

   `certutil -L -d certificate_database_directory`

   -L

   Lists all of the certificates in the certificate database.

   -d *certificate_database_directory*

   Specifies the path to the directory that contains the certificate database.

   Note: If the file path contains spaces, bracket the path in quotes.

   displays the root CA alias, the server certificate alias, and the trust attributes you specified when adding the certificates to the certificate database.

Example: List the Certificates in the Certificate Database

```
certutil -L -d C:\certdatabase
```

SSL-enable the LDAP User Directory Connection

After pointing the system to the correct certificate database, enable the SSL-secured connection to the LDAP user directory. SSL further secures the communication between the Policy Server and the user directory.

Note: The following procedure assumes that you have an LDAP connection working properly.

Follow these steps:

1. Log in to the Administrative UI.
2. Select the User Directory tab.

   The User Directory List is displayed.

3. Click Action, Modify next to the LDAP entry you want to SSL-enable.
4. Verify that the Server field in the Configure LDAP User Directory section contains the correct server and port value for the SSL connection. SSL often uses a different port than a non-SSL connection.
5. Select the Secured Connection check-box in the Connection Credentials section.

6. Click Save.

   You return to the User Directory dialog.

7. In the User Directory list, select Action, Test Connection next to the LDAP entry that is SSL-enabled.

   A message at the top of the dialog either confirms that the SSL is properly configured reports and error.

The user directory connection is configured to communicate over SSL.

Establish a Connection to the Certificate Database

Connecting to an LDAP user directory over SSL requires that the system point to the proper certificate database. This database must contain the cert8.db and key3.db files.

The XPSConfig tool, which is shipped with the product, enables you to specify the path to the certificate database using the LdapObjCertDbPath setting.

Follow these steps:

1. Open a command window.
2. Navigate to *federation_install_dir*.
3. Enter XPSConfig. The command is case-sensitive on UNIX platforms.
4. Enter SM.
5. Enter the number for the LdapObjCertDbPath setting.
6. Enter C to change the value.
7. Specify the path to the certificate database for the Enter New Value prompt.

   Example:

   ```
   C:\Program Files\CA\Federation Standalone\ldaps\certdb
   ```

8. Enter Q until you exit from XPSConfig.

   The new value is saved.

The correct certificate database is now in use.

Verify the SSL Connection to the LDAP Directory

Verify the SSL connection and so you are sure that the user directory connection is secured.

Follow these steps:

1. Log in to the Administrative UI.
2. Select User Directory.

   The User Directories screen appears. The table lists the names of existing user directory connections.
3. Select Action, Modify next to the name of the user directory you want to test.

   The directory settings display.
4. Click View Contents.

   If SSL is properly configured, the Directory Contents screen appears and lists the contents of the user directory.

Troubleshoot the SSL Connection to the LDAP User Directory

The list following specifies actions you can take when you encounter problems connecting to the LDAP user directory using SSL:

- Verify that you can connect to the user directory without using a secured connection.
- Verify that SSL is enabled for the LDAP server you are using.
- Verify that the SSL port of the LDAP server is reachable from the CA SiteMinder® Federation Standalone host.
- Verify that the system is pointing to the directory that contains the certificate database files.
- Verify that that the certificate database directory contains the cert8.db and the key3.db files.
- Verify that the LDAP server configuration (including port), the connection credentials and the search root are configured correctly in the Administrative UI.

ODBC Directory Connection

You can configure a directory connection to an existing ODBC user store (SQL or Oracle) so CA SiteMinder® Federation Standalone can use it for authentication.

Note: If you plan to connect to an ODBC data source on Solaris, configure the wire protocol driver for the data source. See the wire protocol driver instructions for details.

Follow these steps:

1. Click the User Directory tab.
2. Click Connect to ODBC in the User Directory List section.
3. Configure the settings in this dialog. Parameters marked by red dots are required.

   Note: Click Help for a description of fields, controls, and their respective requirements.
4. Click Failover if you want to set up additional ODBC directories for redundancy.
5. Click Test Connection to validate the connection.

   You can click View Contents to list the contents of the user directory.

   Note: The View Contents button is displayed only if the Universal ID Column value is set.
6. Click Save.

   If your settings are valid, you are redirected to the View User Directories dialog.

   The connection to the ODBC directory is configured.

More Information:

ODBC Data Source on Solaris Configuration Requirement (see page 87)
Configure the Oracle Wire Protocol Driver (see page 87)
Configure the SQL Server Wire Protocol Driver (see page 88)

ODBC Directory Failover Configuration

CA SiteMinder® Federation Standalone can distribute ODBC user directory requests over multiple data source servers for failover.

Note: CA SiteMinder® Federation Standalone does not support load balancing for ODBC user directories.

For failover, CA SiteMinder® Federation Standalone uses one ODBC directory to fulfill requests until that server where that store resides fails to respond. When the default directory does not respond, CA SiteMinder® Federation Standalone routes the request to the next store configured for failover. This process can be repeated over multiple servers. After the default server is able to fulfill requests again, CA SiteMinder® Federation Standalone routes requests back to the original server.

To configure ODBC failover

1. Select the User Directory tab in the UI.
2. Do one of the following:
   - Select Connect to ODBC to create an ODBC user directory connection.
   - Select Action, Modify next to an existing ODBC entry you want to edit.

   The User Directory dialog opens.
3. Click Configure Failover in the Configure ODBC User Directory section of the dialog.

   The ODBC Data Source Failover table displays.
4. Enter the data source name, in the first Failover Node field. Add the names of other data sources in the remaining fields for failover.

   Note: If you are adding a server for failover, the failover directory must use the same type of communication (SSL or non-SSL) as the primary directory. Both directories share the same port number.

   If you only have one entry in the table, then CA SiteMinder® Federation Standalone only supports failover.

Example: ODBC Failover

In this example, a CA SiteMinder® environment contains two user directories, A and B, which must meet the following requirements:

- User directory A must failover to user directory B
- User directory B must failover to user directory A

This configuration requires two failover nodes: the data source name for user directory A and a data source name for user directory B.

ODBC Data Source on Solaris Configuration Requirement

If you are using an ODBC data source on a UNIX system as a user directory, configure the data source in the system_odbc.ini file.

The system_odbc.ini file, located in the *federation_install_dir*/siteminder/db folder, contains all of the names of the available data sources. In addition, this file contains the attributes that are associated with these data sources. The first attribute is the ODBC driver allocated to CA SiteMinder® Federation Standalone. The remaining attributes are specific to the driver.

When you are updating the file to configure a new data source, you add a new section that describes the data source. You place entries for the SQL Server or Oracle drivers after the section that reads [CA FedManager Data Source]. Do not modify the original text.

Configure the Oracle Wire Protocol Driver

You configure the Oracle wire protocol driver to specify the settings CA SiteMinder® Federation Standalone uses to connect to the data source.

To configure the Oracle wire protocol driver

1. Navigate to the directory *federation_install_dir*/siteminder/db.
2. Open the system_odbc.ini file in a text editor.
3. Select the section [CA FedManager Data Source] and make a copy of it directly under its current location.
4. Using the copy you created as a template, rename the heading in brackets to something appropriate for your data source.

5. Change the values in the LogonID, Password, HostName, and Service Name entries.

   The modified text for the Oracle data source appears as follows:

   ```
   Driver=federation_install_dir/siteminder/odbc/lib/NSora23.so
   Description=DataDirect 5.3 Oracle Wire Protocol
   LogonID=uid
   Password=pwd
   HostName=servername
   PortNumber=1521
   ServiceName=servicename
   CatalogOptions=0
   ProcedureRetResults=0
   EnableDescribeParam=0
   EnableStaticCursorsForLongData=0
   ApplicationUsingThreads=1
   ```

6. Save the file.

   The Oracle wire protocol driver is configured.

Important! Do not modify other entries in the file, especially anything listed in the [CA FedManager Data Source] section.

Configure the SQL Server Wire Protocol Driver

You configure the SQL wire protocol driver to specify the settings CA SiteMinder® Federation Standalone uses to connect to the database.

To configure the SQL Server wire protocol driver

1. Navigate to the directory *federation_install_dir*/siteminder/db.
2. Open the system_odbc.ini file in a text editor.
3. Select the section [CA FedManager Data Source] and make a copy of it directly under its current location.
4. Using the copy you created as a template, rename the heading in brackets to something appropriate for your data source.
5. Change the values and add new entries so that the modified text for the SQL Server data source appears as follows:

   ```
   Driver=federation_install_dir/siteminder/odbc/lib/NSmass23.so
   Description=DataDirect 5.0 SQL Server Wire Protocol
   Database=database_instance
   Address=host_IP_address, port_number (default: 1433)
   QuotedId=No
   AnsiNPW=No
   ```

6. Save the file.

   The wire protocol driver is configured.

Important! Do not modify other settings in this file, especially anything listed in the [CA FedManager Data Source] section.

Test a User Directory Connection from the Directory List

You can test the connection to a user directory.

To test a user directory connection

1. Click the User Directory tab.

   The View User Directories list is displayed.

2. Select Test Connection from the Actions drop-down menu next to an entry in the list that you want to test.

A message at the top of the dialog verifies the connection or displays an error.

Note: You can also test the connection by clicking Test Connection in the Connection Credentials section of the Create or Modify User Directory dialog.

Create a Common View of the Same User Information Across Directories

Directory connections resolve how CA SiteMinder® Federation Standalone establishes a context for user identities. The asserting party determines which users it can create assertions for by authenticating each user against a user directory.

Multiple user directories in a federated environment often store the same type of user information, but each directory uses a different underlying schema and different user attribute names to identify the information. Therefore, CA SiteMinder® Federation Standalone receives a disparate view of the same user information. For example, an LDAP directory can use the attribute **uid** to represent a user name, whereas an ODBC directory can use the attribute **name** for the same information.

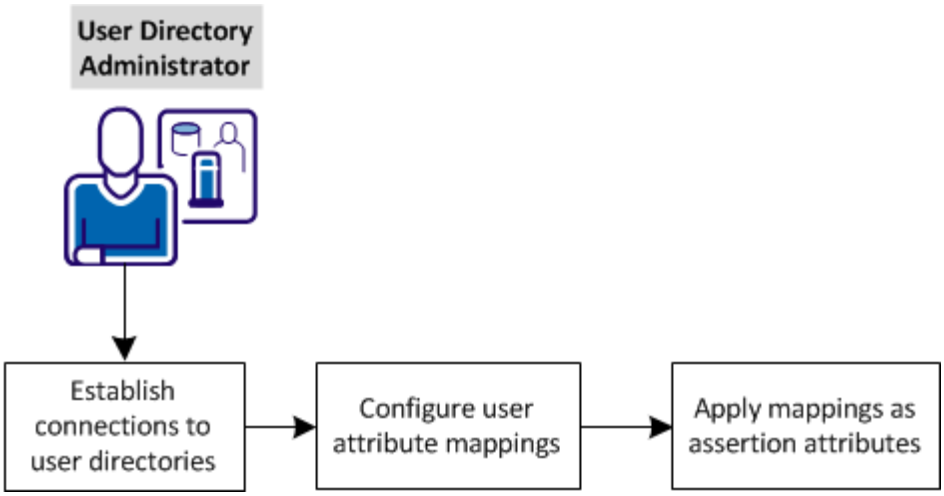
The purpose of user attribute mapping is to create a common view of the same information by defining a universal schema. The universal schema can resolve user information across multiple user directories. The system can reference user attributes without regard for the directory type, greatly reducing the number of configuration objects that are required for multiple user directories.

Each user attribute mapping is specific to the user directory in which it is defined.

After the connections to the user directories are configured, use one common name to reference the same information in different user directories.

The feature that you use to create a universal schema is called *user attribute mapping*. Configure this feature within the user directory configuration of the Administrative UI.

The following graphic shows the process for configuring user attribute mapping at the asserting party.



Complete the following tasks at the asserting party for user attribute mapping:

1. Establish connections to user directories (see page 90).
2. Configure user attribute mappings (see page 91).
3. Apply mappings as assertion attributes (see page 105).

Establish Connections to User Directories

Before you can establish user attribute mappings, establish connections to the user directories that store user records.

LDAP or ODBC are the two types of directories to which the product can connect.

Follow these steps:

1. Click the User Directory tab.
2. Click Connect to LDAP or ODBC.

3. Configure the settings in each section. Required parameters are marked by red dots.

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

4. Click Failover or Load Balancing if you want to set up either of these features.
5. Click Test Connection to verify that the connection is valid.

   You can click View Contents to list the contents of the user directory.

   Note:

   - For an LDAP directory connection, the View Contents button is displayed only if the Search Root, Start User DN Lookup, End User DN Lookup, and Universal ID Attribute values are set.
   - For an ODBC directory connection, the View Contents button is displayed only if the Universal ID Column value is set.

6. Click Save.

   If your settings are valid, you are redirected to the View User Directories dialog.

   The connection to the directory is configured.

Configure User Attribute Mappings

Use one or more of the following mapping types to define attribute mappings:

- Alias
- Group Name
- Mask
- Constant
- Expression

The following table lists the type of data you can enter in the mapping definition. Define individual mappings for each user directory in your deployment.

Mapping Type	Map Common Name to...	Data Types	Access
Alias	A user attribute name in the directory.	String, Number, Boolean	Read/Write
Group name	An attribute that identifies whether a user belongs to a specific group.	Boolean	Read/Write
Mask	A user attribute that stores a bit pattern.	Boolean	Read/Write

Mapping Type	Map Common Name to...	Data Types	Access
Constant	A value that is the same or *constant* for every user in a directory.	String, Number, Boolean	Read
Expression	To an expression. For complete syntax information, see the Attributes and Expression Reference appendix in the *SiteMinder Policy Server Configuration Guide.* This guide is part of the SiteMinder bookshelf.	String, Number, Boolean	Read

The configuration procedures for each mapping type are basically the same. Refer to the use cases for each mapping type for implementation examples.

Follow these steps:

1. In the Administrative UI, navigate to the User Directory tab.
2. Select one of the *Connect to* options in the User Directory List.
3. Verify that a user directory connection is configured or configure one.
4. Scroll to the Directory Mapping Attribute section and select Create Mapping.
5. Complete the General fields:

   Name

   Specify the common name for this mapping. Common names must conform to the same rules as user attribute names.

   Description

   Enter a description of the attribute mapping.
6. Complete the Properties fields:

   Mapping type

   Select the mapping type that you want to configure.

   Definition

   Enter the mapping definition using the appropriate syntax. Refer to the previous table.
7. (Optional) Select Disabled to disable an attribute mapping.
8. Click Save.

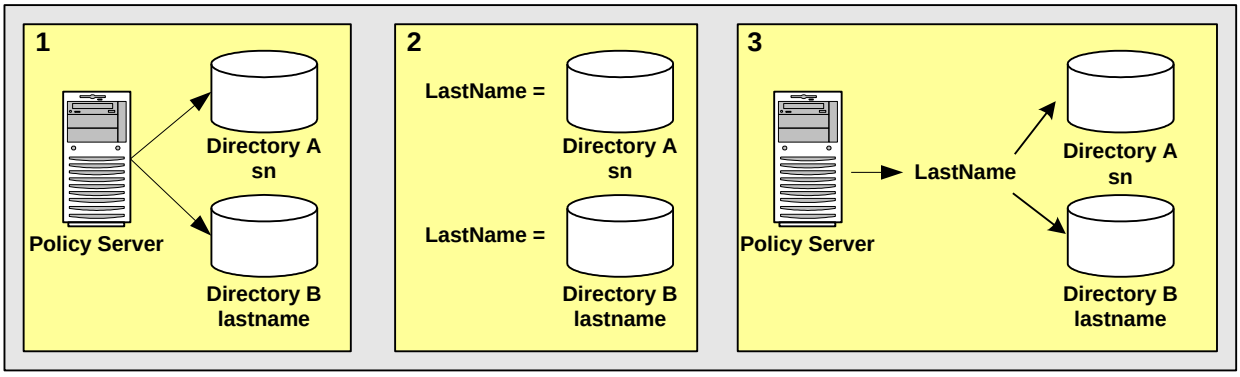
The new attribute mapping is submitted and then added to the list on the Attribute Mapping List table.

Alias Attribute Use Case

This use case shows two LDAP user directories, which identify the last name of users, but the directories have different underlying schema.

Note: Review the advanced user attribute mapping examples, which detail how to use different attribute mapping types to identify the same user attribute across different directory types.

The following illustration details how two alias attribute mappings can create a common view of the same user information.



1. Two user directories identify the last name of users differently:
   - Directory A identifies the last name of users with sn.
   - Directory B identifies the last name of users with lastname.

   This results in two different views of the same user information.
2. LastName is the common name or *alias* that is mapped to the underlying directory schema:
   - LastName is mapped to sn in Directory A.
   - LastName is mapped to lastname in Directory B.

LastName results in a common view of the same user information. Use LastName when defining assertion attributes or NameID attributes that use the last names. The system has no concern for the directory-specific schema because the directories are operationally identical.

More information:

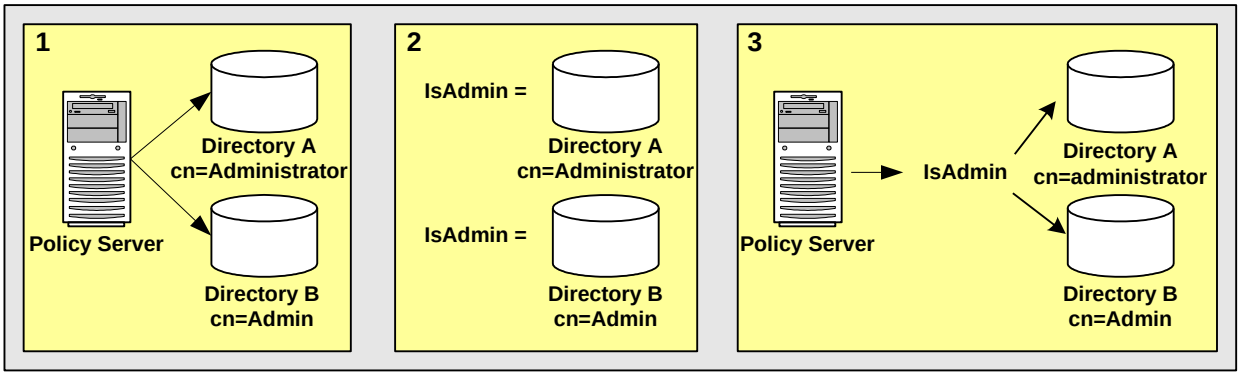
Advanced User Attribute Mapping Examples (see page 100)

Group Name Use Case

This use case shows two LDAP user directories, which use different underlying schema to identify users that belong to an Administrator group.

Note: Review the advanced user attribute mapping examples, which detail how to use different attribute mapping types to identify the same user attribute across different directory types.

The following illustration details how two group name attribute mappings can create a common view of the same user information.



1. Two user directories identify membership to the administrator group differently:
   - Directory A identifies membership in the administrator group as cn=Administrators,ou=groups,o=acme.com.
   - Directory B identifies membership in the administrator group as cn=Admin,ou=groups,o=acme.com.

   This results in two different views of the same user information.
2. IsAdmin is the common name that is mapped to the underlying directory schema:
   - IsAdmin is mapped to cn=Administrators,ou=groups,o=acme.com in Directory A.
   - IsAdmin is mapped to cn=Admin,ou=group,o=acme.com in Directory B.

IsAdmin results in a common view of the administrator group. You can reference IsAdmin when defining assertion attributes or NameID attributes that apply to the Administrator group. The system has no concern for the directory-specific schema because the directories are operationally identical.

More information:

Advanced User Attribute Mapping Examples (see page 100)

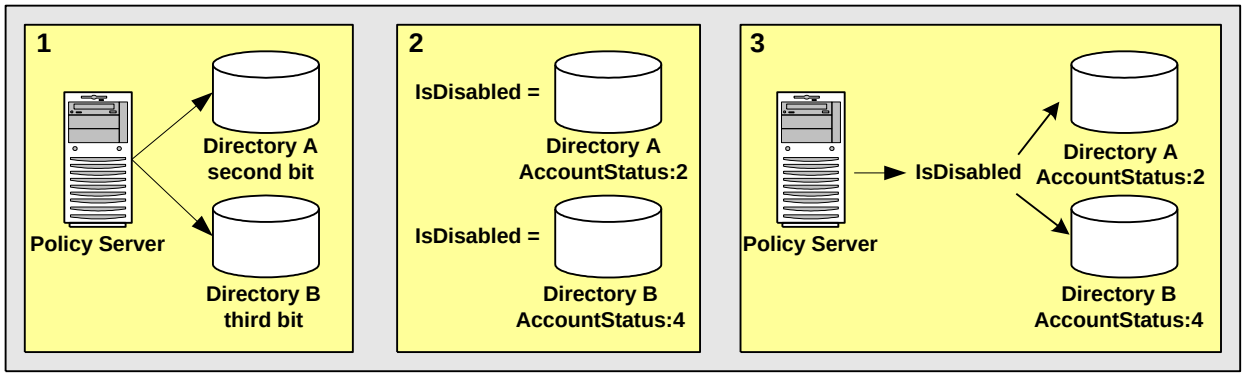
Mask Use Case

Some directory implementations use individual bits in an attribute to provide information about that attribute, such as the state of an account. You can apply a bit mask to an attribute.

This use case shows two Active Directory user stores that identify disabled user accounts. Each account has a different underlying schema.

Note: Review the advanced user attribute mapping examples, which detail how to use different attribute mapping types to identify the same user attribute across different directory types.

The following illustration details how two mask attribute mappings can create a common view of the same user information.



1. Two user directories contain a user attribute named AccountStatus. AccountStatus stores user information in a bit pattern, where each bit is a flag.
   - In Directory A, the second bit flags a disabled account. When the second bit equals 1, the account is disabled.
   - In Directory B, the third bit flags a disabled account. When the third bit equals 1, the account is disabled.

   This results in two different views of the same user information.
2. IsDisabled is the common name that is mapped to the underlying directory schema. In both directories, IsDisabled is mapped to AccountStatus.
   - In Directory A, the bit mask 2 (decimal) determines whether the second bit of AccountStatus is set and the account is disabled.
   - In Directory B, bit mask 4 (decimal) determines whether the third bit of AccountStatus is set and the account is disabled.

IsDisabled results in a common view of disabled user accounts. You can reference IsDisabled when defining assertion attributes or NameID attributes that require the account status of users. The system has no concern for the directory-specific schema because the directories are operationally identical.

More information:

Advanced User Attribute Mapping Examples (see page 100)

Bit Masks in Mask Attribute Mapping

A bit mask attribute mapping tests the value of one or more bits by masking the values of the other bits in a user attribute.

A mask attribute mapping is defined as follows:

```
user_attribute_name:bit_mask
```

For example, assume that the user attribute is named AccountStatus. The attribute AccountStatus stores the states of the following three flags in a bit pattern:

Bit Pattern	Flag
00?	account disabled?
0?0	password expired?
?00	gold member?

When a bit equals one, the flag is TRUE. The table shows the results:

Bit Pattern	Account Status
000 (0)	no flags are TRUE
001 (1)	account disabled
010 (2)	password expired
100 (4)	gold member
011 (3)	password expired, account disabled
101 (5)	gold member, account disabled
110 (6)	gold member, password expired
111 (7)	gold member, password expired, account disabled

Note: Equivalent decimal values are shown in parentheses.

Assume that you only want to test whether a user is a gold member. To test this bit, select the bit pattern that corresponds to a gold member as the bit mask or 100 (binary) and specify it as 4 (decimal). The resulting mask attribute mapping is defined as follows:

```
AccountStatus:4
```

A bitwise AND operation on AccountStatus is performed on the bit mask and tests whether the result is equal to the bit mask. An equal result means the value of the tested bit is one and the flag is TRUE. The following table shows the results:

Account Status	Bit Mask	Result of Bitwise AND	Gold Member?
000 (0)	100 (4)	000 (0)	FALSE
001 (1)	100 (4)	000 (0)	FALSE
010 (2)	100 (4)	000 (0)	FALSE
011 (3)	100 (4)	000 (0)	FALSE
100 (4)	100 (4)	100 (4)	TRUE
101 (5)	100 (4)	100 (4)	TRUE
110 (6)	100 (4)	100 (4)	TRUE
111 (7)	100 (4)	100 (4)	TRUE

Note: Equivalent decimal values are shown in parentheses.

You can also use a bit mask to test the value of a bit set or more than one bit at a time. Assume that you want to know whether the account is disabled and the password has expired. To test these bits, specify a bit mask of 011 (binary) or 3 (decimal). The resulting mask attribute mapping is defined as follows:

```
AccountStatus:3
```

A bitwise AND operation on AccountStatus is performed on the bit mask and tests whether the result is equal to the bit mask. An equal result means the value of both tested bits is one and both flags are TRUE. The following table shows the results:

Account Status	Bit Mask	Result of Bitwise AND	Both Flags Set?
000 (0)	011 (3)	000 (0)	FALSE
001 (1)	011 (3)	001 (1)	FALSE
010 (2)	011 (3)	010 (2)	FALSE
011 (3)	011 (3)	011 (3)	TRUE
100 (4)	011 (3)	000 (0)	FALSE

Account Status	Bit Mask	Result of Bitwise AND	Both Flags Set?
101 (5)	011 (3)	001 (1)	FALSE
110 (6)	011 (3)	010 (2)	FALSE
111 (7)	011 (3)	011 (3)	TRUE

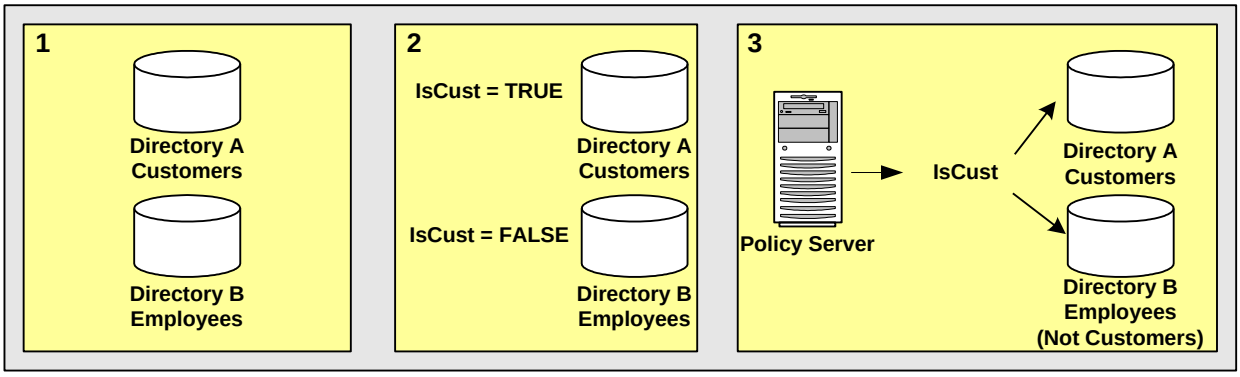
Note: Equivalent decimal values are shown in parentheses.

Constant Use Case

This use case represents a scenario in which one user directory stores only customers, while another user directory stores only employees.

Note: Review the advanced user attribute mapping examples, which detail how to use different attribute mapping types to identify the same user attribute across different directory types.

The following illustration details how two constant attribute mappings can represent different values for different user directories.



1. Directory A only stores customers. Directory B only stores employees.
2. IsCust is the common name that is mapped to different values in different directories:
   - IsCust is mapped to TRUE in Directory A.
   - IsCust is mapped to FALSE in Directory B.
3. Reference IsCust when defining assertion attributes or NameID attributes. The common name lets the system determine whether a user is a customer, without regard to the particular directory in which the user is stored. The mapping indicates that every user in Directory A is a customer, while every user in Directory B is not a customer.

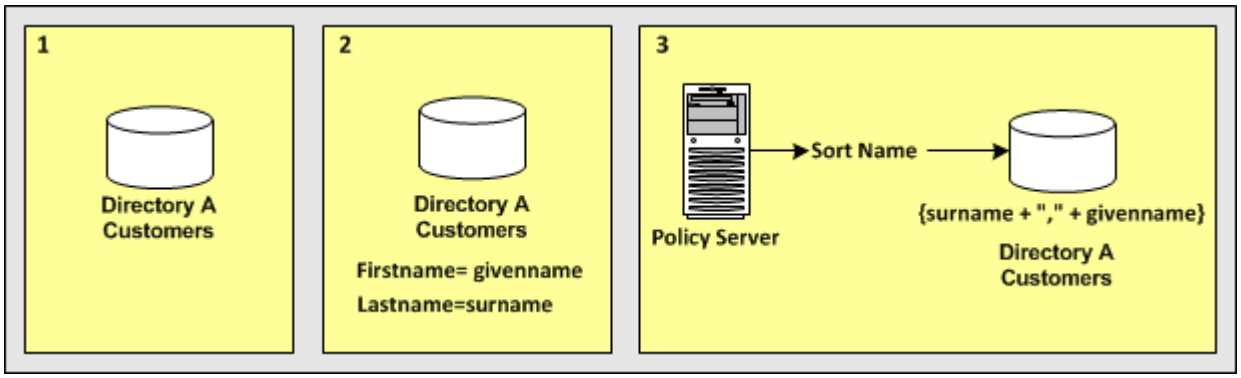
More information:

Advanced User Attribute Mapping Examples (see page 100)

Expression Use Case

This use case shows how you can use an expression attribute mapping to simplify references to multiple user attributes in one directory. A protected resource needs the *sort name* of each user (last name,first name). The user directory does not uniquely reference this attribute. Instead, the directory does store the last name of each user as surname and the first name of each user as givenname.

The following illustration details how an expression attribute mapping can create a common view of the same user information.



In the single user directory, a common name is mapped to an expression that creates the sort name using the user attribute names in the directory.

- Directory A contains all user records.
- The name of the mapping is **SortName**.
- The expression that defines SortName is:

  ```
  {surname + "," + givenname}
  ```

  Note: The expression conforms to the syntax rules of a CA SiteMinder® expression. For complete syntax information, see the Attributes and Expression Reference appendix in the *SiteMinder Policy Server Configuration Guide in the* SiteMinder bookshelf.
- SortName is the common name that is mapped to the expression that includes the surname and the givenname attributes.

Reference SortName when defining assertion attributes or NameID attributes that require the sort name of users without concern for the directory-specific schema.

More information:

Advanced User Attribute Mapping Examples (see page 100)

Advanced User Attribute Mapping Examples

The following examples show more complex user attribute mapping configurations.

The example deployment is a retail clothing company that uses two user directories of different types:

Directory A

An internal LDAP user directory for employees only.

Directory B

An ODBC user directory for customers only.

Each user attribute mapping is specific to the user directory for which it is defined.

The following table details how Directory A and Directory B identify the same user information. The accompanying use cases explain how to use different attribute mappings to define a common view of the same user information. The common view serves as a universal schema, which makes the directories operationally identical.

Attribute Description	Directory A Attributes (LDAP)	Directory B Attributes (ODBC)
First name of each user	givenname	u_first_name
Last name of each user	surname	u_last_name
Sort name of each user (last name, first name)	The user directory does not uniquely store the user attribute.	sort_name
User as a customer	group:cn=customer,ou=groups,o=acme.com	Users are always customers.
Status of a user account	AccountStatus attribute (a set of flags). Second bit is a disabled account.	u_disabled

Map a First Name Attribute with an Alias Mapping Type

Use two alias attribute mappings to represent the first name user attribute in Directory A and Directory B.

Deployment

User Directory A identifies the first name of users with givenname. Directory B identifies the first name of users with u_first_name.

Solution

1. Create an alias attribute mapping for Directory A.

   Name

   FirstName

   Mapping Type

   Alias

   Definition

   givenname

2. Create an alias attribute mapping for Directory B.

   Name

   FirstName

   Mapping Type

   Alias

   Definition

   u_first_name

When referencing users in Directory A, the FirstName is mapped to givenname. When referencing users in Directory B, the FirstName maps to u_first_name.

Map a Last Name Attribute with an Alias Mapping Type

Use two alias attribute mappings to represent the last name user attribute in Directory A and Directory B.

Deployment

User Directory A identifies the last name of users with surname. Directory B identifies the last name of users with u_last_name.

Solution

1. Create an alias attribute mapping for Directory A.

   Name

   LastName

   Mapping Type

   Alias

   Definition

   surname

2. Create an alias attribute mapping for Directory B.

   Name

   LastName

   Mapping Type

   Alias

   Definition

   u_last_name

When referencing users in Directory A, the common view determines that the last name of users is identified by surname. When referencing users in Directory B, the common view determines that the last name is identified by u_last_name.

Map a Sort Name Attribute with Expression and Alias Mapping Types

Use an expression attribute mapping and an alias attribute mapping to represent the sort name of a user in Directory A and Directory B.

Deployment

- Directory A does not uniquely identify a the sort name for each user. For each user, Directory A stores the first name as givenname and a last name as surname for each user.
- Directory B identifies a sort name using sort_name.

Solution

1. Create an expression attribute mapping for Directory A:

   Name

   SortName

   Mapping Type

   Expression

   Definition

   ```
   (surname + "," + givenname)
   ```

   Note: The expression must conform to the syntax rules of an expression.

2. Create an alias attribute mapping for Directory B:

   Name

   SortName

   Mapping Type

   Alias

   Definition

   sort_name

When referencing users in Directory A, the sort name is calculated based on the specified expression. When referencing users in Directory B, the sort name is represented by the attribute sort_name.

Map Customers with Group and Constant Mapping Types

Use a group and a constant attribute mapping to identify customers in Directory A and Directory B.

Deployment

- Directory A stores employee. An employee of the company can also be a customer, so Directory A identifies customers as those employees that belong to the following group:

  `cn=Customers,ou=Groups,o=acme.com`

- Directory B only stores customers. Directory B does not have a user attribute that identifies customers because to store a user in Directory B implies that the user is a customer.

Solution

1. Create a group attribute mapping for Directory A.

   Name

   IsCustomer

   Mapping Type

   Group

   Definition

   `cn=Customers,ou=Groups,o=acme.com`

2. Create a constant attribute mapping for Directory B.

   Name

   IsCustomer

   Mapping Type

   Constant

   Definition

   TRUE

When referencing Directory A, a user is considered a customer if they belong to cn=Customers,ou=Groups,o=acme.com. When referencing Directory B, every user is a customer.

Map the Account Status with the Mask and Expression Mapping Types

Use a mask attribute mapping and an expression attribute mapping to identify user accounts that are disabled in Directory A and Directory B.

Deployment

- Directory A identifies disabled accounts with a user attribute named AccountStatus, which is a set of flags. The second bit indicates a disabled account.
- Directory B identifies disabled accounts with a user attribute named u_disabled. When u_disabled is equal to "y", the account is disabled. When u_disabled is equal to "n", the account is active.

Solution

1. Create a mask attribute mapping for Directory A.

   Name

   IsDisabled

   Mapping Type

   Mask

   Definition

   AccountStatus:2

   The definition indicates that the bit pattern is stored in AccountStatus, and the bit mask is 2 (decimal).

2. Create a expression attribute mapping for Directory B.

   Name

   IsDisabled

   Mapping Type

   Expression

   Definition

   ```
   (u_disabled = "y")
   ```

   u_disabled is a Boolean expression.

When referencing Directory A, the bit pattern determines if a user is disabled. When referencing Directory B, the expression determines if a user is disabled.

Apply Mappings to Assertion Attributes

After you define user attribute mappings for your user directories, add the user attribute mapping to the assertion configuration for the asserting-to-relying party partnership. The mapping helps the asserting party include the right attributes in the assertion, regardless of the different attributes for each directory type.

The Name ID type can be a user attribute in the assertion configuration.

Follow these steps:

1. Log on to the Administrative UI.
2. Click Federation, Partnerships.
3. Select Action, Modify for the local asserting party partnership.
4. Navigate to the Assertion Configuration tab.
5. In the Assertion Attributes section, click Add Row.
6. Enter the data from the user mappings into the fields as follows:

   Assertion Attribute

   Specify any name for the name/value pair of an assertion attribute.

   Format

   Choose the format that indicates how to interpret the attribute name.

Type

User attribute

Always select the user attribute type as the value for this field.

Value

Enter the value from the Name field in the user mapping section of the User Directory dialog.

Example: If the Name you assigned to a mapping is FullName, enter FullName in this field.

7. (Optional). The Name ID type can be a user attribute so make the Value field for the Name ID entry match the Value field in the assertion attribute entry. The assertion then uses the same user attribute for the Name ID and the assertion attribute that identifies the user.
8. Repeat the procedure in the previous step for all assertion attributes.
9. Navigate to the Confirm step and click Finish to save your changes.

Chapter 6: Federation Entity Configuration

This section contains the following topics:

Methods to Create an Entity (see page 107)
How to Create an Entity without Using Metadata (see page 107)
How to Create an Entity by Importing Metadata (see page 111)

Methods to Create an Entity

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

The two ways to configure a federation entity are:

- Create an entity without using metadata.
- Create an entity by importing metadata.

How to Create an Entity without Using Metadata

Create an entity without metadata by using the following process:

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

Entity Type Choice

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

Follow these steps:

1. Log in to the Administrative UI.
2. From the Federation tab, select Entities.
3. Click Create Entity.

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

4. Select one of the following options for the entity location:

   Local

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

   Remote

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

5. Identify the specific entity type in the New Entity Type field. The drop-down list shows all the options.
6. Click Next to configure specifics about the entity.

Detailed Local Entity Configuration

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

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

Be aware of the following concepts:

Entity ID and Entity Name Settings

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

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

Note: The Entity Name can be the same value as the Entity ID, but the value cannot be shared.

Signing and Encryption Features

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

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

Assertion Attributes Configuration

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

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

Follow these steps:

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

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

   The Confirm dialog is displayed.

Detailed Remote Entity Configuration

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

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

Be aware of the following concepts:

Entity ID and Entity Name Settings

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

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

Note: The Entity Name can be the same value as the Entity ID, but the value cannot be shared.

Signing and Encryption Features

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

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

Assertion Attributes Configuration

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

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

Follow these steps:

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

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

The Confirm dialog is displayed.

Confirm the Entity Configuration

Review the entity configuration before saving it.

Follow these steps:

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

A new entity is configured.

Entity Configuration Changes from a Partnership

You can change an entity ID value for the remote entity from within the context of a single partnership configuration. However, changing the entity ID at the partnership level does not link the partnership to another entity, nor does it update the original entity. Modifications to an entity are a one-way propagation from the entity to the partnership. A change to the entity ID at the partnership level does not get propagated to the original entity.

Note: The entity ID you specify has to match what your remote partner is using.

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

Refer to editing an entity from a partnership (see page 146) for more details about entities within a partnership.

How to Create an Entity by Importing Metadata

Import data from a metadata file to create a federation entity. Importing SAML metadata reduces the amount of configuration that is required to create a partnership.

You can use metadata in the following ways:

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

  This option is useful to facilitate a migration to CA SiteMinder® Federation Standalone from another federation product.

Note: The product does not support metadata imports to update or restore an existing partnership and local entity. To update an existing local entity, edit the entity and modify the settings that require change. Import metadata only to create a *new* local entity.

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

1. Select a metadata file to serve as the basis for configuring a new entity.
2. Select an entity entry from the metadata file. The file can include several entities, but one entity per file is recommended.

3. (Optional) To configure an entity, select certificates in the metadata file to import.

   These certificates are for various federation functions, such as signing, verification, or single logout.

4. Confirm the entity configuration.

Details about these steps are described in the next sections.

Metadata File Selection

The first step to create a federation entity based on metadata is to select the metadata file.

Follow these steps:

1. Log in to the Administrative UI.
2. From the Federation tab, select Entities.
3. Click Import Metadata.

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

4. Browse for the metadata file you want to use to create the entity.
5. Select whether to create a new local or remote entity, or update an existing remote entity.

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

6. Click Next to select entities from the file.

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

Select an Entity to Import

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

Follow these steps:

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

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

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

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

3. Click Next.

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

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

Certificate Imports

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

Follow these steps:

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

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

   Specify a unique alias for each entry that you chose.

2. Click Next

   The Confirm dialog displays showing a table of entries.

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

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

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

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

Alias	Issued To	Usage
cert1	Jane Doe	Signing and Encryption

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

3. Click Next to finish the configuration.

Confirm the Entity Configuration

Review the entity configuration before saving it.

Follow these steps:

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

A new entity is configured.

Chapter 7: Key and Certificate Management

This section contains the following topics:

Certificate and Private Key Usage (see page 115)
Obtain a Key/Certificate Pair for Federated Transactions (see page 119)
How to Verify that Certificates are Valid Using CRLs (see page 124)
How to Verify that Certificates are Valid using OCSP (see page 129)
How to Send Certificates to Your Partner (see page 132)
Update Certificates in the Certificate Data Store (see page 138)
Certificate Authority (CA) Certificate Usage (see page 139)

Certificate and Private Key Usage

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 overview 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.

Aliases to Reference Certificate Data Store Content

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

If a signing alias is configured for signing assertions, the assertion generator uses the key associated with alias to sign assertions. If no signing alias is configured, the assertion generator uses the key with the following alias to sign assertions:

```
defaultenterpriseprivatekey
```

If the assertion generator does not find a default enterprise private key, it uses the first private key in the store to sign assertions.

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

```
defaultenterpriseprivatekey
```

A given Policy Server signs or signs and verifies responses. Add keys and certificates for signing and validation to the same certificate data store.

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

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

Signing and Verification Operations

The system uses a private key/certificate pair for signing and verification tasks. The private key/certificate pair signs the assertion, the assertion response, or authentication request, depending on the transaction taking place. Before any signing transaction, the partner signing the assertion sends the certificate (public key) associated with the private key/certificate pair to the partner. This exchange is done as part of an out-of-band communication. The partner uses the certificate to verify the signature.

When a transaction occurs, the asserting party includes the certificate in the assertion, by default. During verification, however, the partner uses the certificate that it stores at its site to validate the signature.

For SAML 2.0 single logout, the side that initiates the logout signs the request, and the side receiving the request validates the signature. Conversely, the receiving side signs the SLO response and the initiator validates the response.

Encryption and Decryption Operations

For SAML 2.0, you can configure CA SiteMinder® Federation Standalone to encrypt an entire assertion, the NameID, or other attributes. If you enable encryption, the asserting party uses the certificate (public key) the relying party sends to encrypt data. Before any transaction, the relying party sends the certificate to the asserting party in an out-of-band exchange. The relying party uses the private key/certificate pair to decrypt the data.

Note: SAML 1.1 and WS-Federation do not support encryption of assertion data.

Certificates for SSL Connections

Enable SSL for the artifact back channel to secure the SSL connection and make back-channel communication secure.

To establish the SSL connection, the relying party has to associate the CA certificate with the signed SSL server certificate. The SSL server certificate secures the SSL connection, while the CA certificate verifies that the SSL server certificate is trusted.

Certificates to Secure the Artifact Back Channel

To implement single sign-on using the artifact binding, the relying party sends a request for an assertion to CA SiteMinder® Federation Standalone at the asserting party. The assertion request goes to the Assertion Retrieval Service (SAML 1.1) or the Artifact Resolution Service (SAML 2.0). The retrieval service takes the artifact supplied by the relying party and uses it to retrieve the assertion. CA SiteMinder® Federation Standalone sends the response back to the relying party over a back channel. The back channel is a secured connection between the asserting and relying party. In contrast, web browser communication occurs over the front channel.

Secure the back channel and the retrieval service from unauthorized access using one of the following authentication methods:

- Basic
- Basic over SSL
- X.509 Client Certificate

  If you use X.509 client certificate as the authentication method, the relying party must provide a client certificate as its credential. This credential lets the replying party gain access to the service at the asserting party that retrieves the assertion.

Consider the following items when choosing an authentication method:

- Consider using an SSL connection for the back channel. Secure the SSL connection with an SSL server certificate signed by a trusted CA.

  A default set of common root and intermediate CA certificates are shipped with the certificate data store. To use another server certificate signed by a CA, import the CA certificate into the store as a trusted CA certificate.

  Federation uses an SSL-client when processing back channel requests. You can configure the web server at the asserting party to use SSL versions TLSV1_1 and TSLV1_2 with the following ciphers:

  - RSA_With_AES_128_CBC_SHA256
  - RSA_With_AES_256_CBC_SHA256

  These ciphers are supported in both FIPS and non-FIPS mode. The determination whether to use SHA256 is made on the SP server side. Federation has no configuration for selecting the algorithm. Administrators must verify that the server at the asserting party is configured appropriately.

- If an X.509 client certificate is required to establish a connection, the relying party must have the key/certificate pair or client certificate authentication fails. Verify that the client certificate exists in the certificate data store at the asserting party. When the relying party sends the request for the assertion, the client certificate serves as the relying party credentials to access the retrieval service.

Obtain a Key/Certificate Pair for Federated Transactions

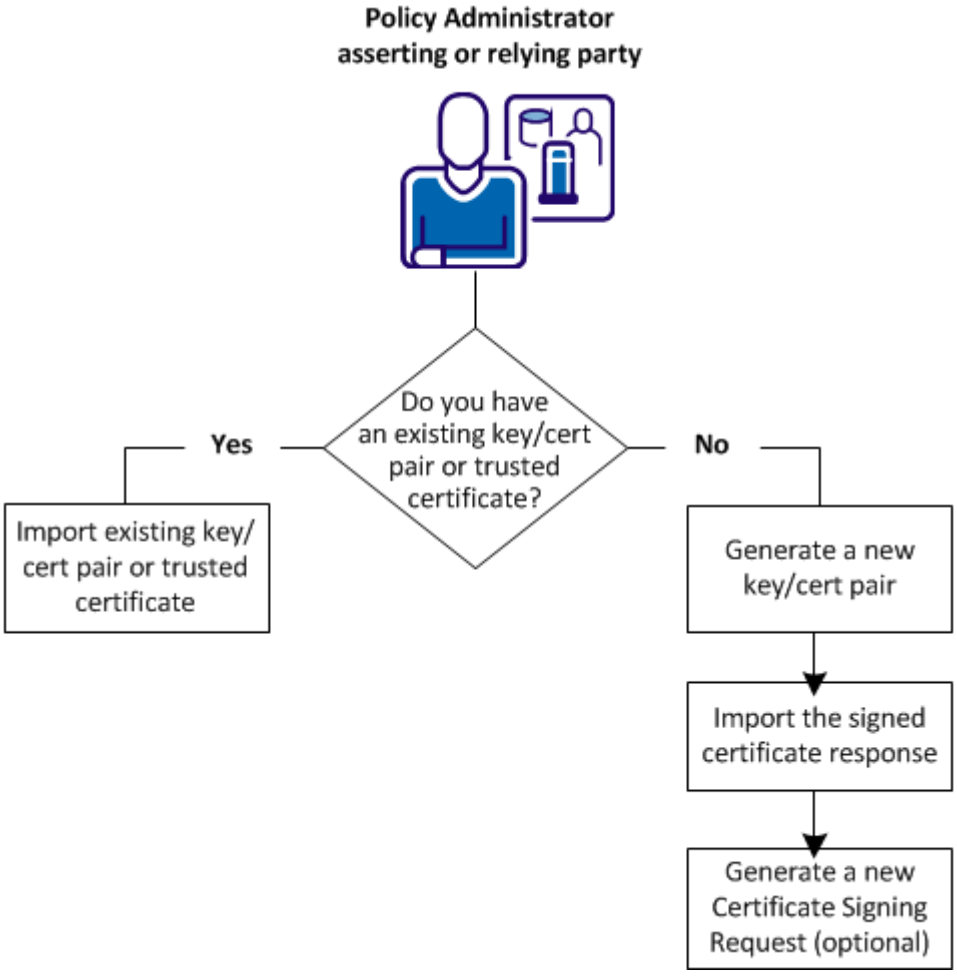
CA SiteMinder® Federation Standalone uses a key/certificate pair and trusted certificates for a number of functions. For CA SiteMinder® Federation Standalone to perform tasks that use keys and certificates, these items must be in the certificate data store.

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

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

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

The following figure shows the steps for each method of obtaining a key/certificate pair or trusted certificate.



Import a Key/Certificate Pair from an Existing File

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

CA SiteMinder® Federation Standalone treats a certificate that you import as a trusted certificate. The exceptions are self-signed certificates:

- If the system identifies a V3 self-signed certificate as a CA certificate, the certificate is treated as a CA certificate. This behavior occurs even though you initiate the import from the Certificate/Private Key dialog.

- CA SiteMinder® Federation Standalone treats the certificate as a trusted certificate:
  - If CA SiteMinder® Federation Standalone does not identify a V3 self–signed certificate as a CA.
  - If the certificate is a V1 self–signed certificate.

Follow these steps:

1. Log in to the Administrative UI.
2. From the Certs & Keys tab, select Certificates and Private Keys.

   The View Certificates and Private Keys dialog opens.
3. Click Import New and follow the wizard.

   Note: You can click Help for a description of fields, controls, and their respective requirements.

   Be aware of the following items as you complete the wizard:

   - You can import a single file with a key and certificate in it or separate key and certificate files. Select the appropriate option button for the file you are using.
   - To import a self-signed certificate as a Certificate Authority certificate, set the Use as CA option button to Yes. The certificate is imported as a CA certificate and is not available for when configuring partnerships (for example, for signing or encryption).

     Otherwise, accept the default No setting to import the certificate as a trusted certificate that is available when configuring partnerships.
   - For a trusted certificate file in DER (binary) format, the file can contain one or more certificate entries. For a trusted certificate file in PEM (base 64) format, CA SiteMinder® Federation Standalone expects one certificate per file.

     The standard extension for a file in DER or PEM format is *.crt or *.cer.
   - If you are using a .p12 file, you are required to fill in a password. CA SiteMinder® Federation Standalone processes a .p12 or .pfx file as a file containing key/certificate pairs.
   - For each entry you plan to add to the certificate data store, enter the alias you want to associate with that entry. If you select multiple entries, each requires a unique alias.
4. At the Confirm step, review the information and click Finish.

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

How to Generate a Key/Certificate Pair

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

Perform the following steps:

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

Generate a Certificate Request

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

When you generate a certificate request, CA SiteMinder® Federation Standalone generates a private key and a self–signed certificate pair. CA SiteMinder® Federation Standalone stores this pair in the certificate data store. Using the generated request, contact a Certificate Authority and fill out the CA certificate request form, pasting the contents of the generated request into the form.

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

Follow these steps:

1. Log in to the Administrative UI.
2. From the Certs & Keys tab, select Certificate and Private Keys.

   The View Certificates and Private Keys dialog opens.
3. Click Request Certificate.

   The Request Certificate dialog opens.
4. Complete the required fields.

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

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

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

Import a Signed Certificate Response

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

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

Follow these steps:

1. From the Certs & Keys tab, select Certificate and Private Keys.

   The View Certificates and Private Keys dialog opens.

2. Search for the self-signed entry with the same alias.

3. Select Action, Update Certificate next to the entry that contains the self-signed certificate.

   The wizard for importing certificates and keys displays.

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

4. Browse to the file you want. You can use a:

   - .p7 or .p7b file that contains the signed certificate and the corresponding certificate chain.
   - .cer or .crt file (base64 PEM file) with the signed certificate without the certificate chain.

5. Select the appropriate entry.

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

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

Generate a New Certificate Signing Request

A certificate signing request (CSR) is a message that you send to a Certificate Authority to apply for a identity certificate. Before you can generate a CSR, CA SiteMinder® Federation Standalone has to generate a key/certificate pair. The certificate is then placed in the CSR.

Generate a new request for an existing private key because:

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

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

Follow these steps:

1. From the Certs & Keys tab, select Certificate and Private Keys.

   The Certificate and Private Key List displays.

2. Select Action, Generate CSR for the private key entry for which you want a new CSR.

   A file that conforms to the PKCS #10 specification is generated and CA SiteMinder® Federation Standalone prompts you to save the CSR.

3. Click Save.

4. (Optional) If you require a CA-signed certificate, contact a Certificate Authority and follow the procedure the Certificate Authority requires for submitting a request. Use the PKCS#10 file you saved in the previous step for the request.

After you complete the certificate request process, the Certificate Authority issues a signed certificate response that you import into the certificate data store. CA SiteMinder® Federation Standalone replaces the existing certificate entry of the same alias with the newly imported certificate.

How to Verify that Certificates are Valid Using CRLs

A Certificate Revocation List (CRL) is issued by a Certificate Authority to its subscribers. The list contains serial numbers of certificates that are invalid or have been revoked. When a request to access a server is received, the server allows or denies access based on the CRL.

CA SiteMinder® Federation Standalone can leverage CRLs for its certificate functions. For CA SiteMinder® Federation Standalone to use a CRL, the certificate data store must point to a current CRL. If CA SiteMinder® Federation Standalone tries using a revoked partner certificate, you see an error message. For legacy federation, the error message is in the SAML assertion. The message indicates that authentication failed.

CA SiteMinder® Federation Standalone supports the following CRL features:

- File-based CRLs or LDAP CRLs

  CA SiteMinder® Federation Standalone stores CRLs in the certificate data store. File-based CRLs must be in Base64 or binary encoding. LDAP CRLs must be in binary encoding. Additionally, LDAP CRLs must include CRL data in one of the following attributes:

  - certificateRevocationList;binary
  - authorityRevocationList;binary

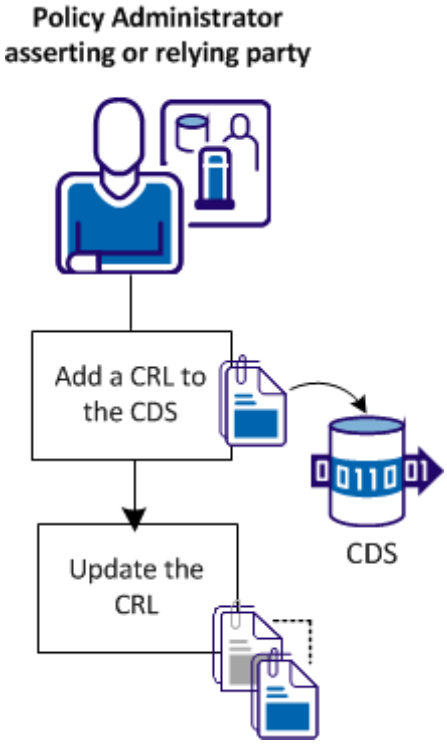
  When a Certificate Authority publishes an LDAP CRL, it must return the CRL data in binary format, in accordance with RFC4522 and RFC4523. Otherwise, CA SiteMinder® Federation Standalone cannot use it.

- PEM and DER encoding formats for a file CRL
- DER encoding format for an LDAP CRL

CA SiteMinder® Federation Standalone does not validate an SSL server certificate against a CRL. The web server where CA SiteMinder® Federation Standalone is installed manages the SSL server certificate.

You are not required to have a CRL for each root CA in the system. If there is no CRL for the root CA, CA SiteMinder® Federation Standalone assumes that all certificates signed by that CA are trusted certificates.

The following figure shows the procedures for managing CRLs.



The CRL configuration steps are as follows:

1. Add a CRL to the CDS. (see page 126)
2. Update a CRL (see page 127).

Add a CRL to the CDS

Ensure that only valid certificates are being used for federation-related PKI functions by using CRLs against which certificates can be checked.

Important! CA SiteMinder® Federation Standalone explicitly requests LDAP CRLs in binary transfer encoding, using the certificateRevocationList;binary LDAP attribute. This means that the CRL data must be stored in this attribute. When a Certificate Authority (CA) publishes a CRL using the LDAP protocol, it must return the CRL data in binary format, in accordance with RFC4522 and RFC4523.

For CA SiteMinder® Federation Standalone to use a CRL, specify the CRL location.

Follow these steps:

1. Go to the Certs and Keys tab.
2. Select the Revocation Lists (CRL).

   The list of available CRL locations is displayed.
3. Click Add.

   The Add Certificate Revocation List is displayed.

   Note: You can click Help for a description of fields, controls, and their respective requirements.
4. Specify an alias for the issuer of the CRL and the location (URL) of the certificate revocation list.

   The location has to be a file path for a file CRL and an LDAP search path for an LDAP CRL.
5. Click Save.

The CRL is now added to the certificate data store.

Update a CRL

Update a CRL to verify that the certificate data in use is current.

Follow these steps:

1. Log in to the Administrative UI.
2. Select the Certs and Keys tab.
3. Select CDS Settings.

   The Certificate Settings dialog displays.
4. Complete one of the following steps
   - If a stored CRL file does not contain a NextUpdate value, set the CRL Update Period to specify the frequency that the next CRL is issued.
   - To change the frequency at which the udpater checks for updates, modify the CRL Updater Sleep Period.

   Note: Click Help for a description of fields, controls, and their respective requirements.
5. In the CRL Updater section, select Enabled in the CRL Updater State field.
6. Click Save.

Manage Certificate Cache Refresh and Grace Period

You can complete two other tasks to manage certificate validity checking (CRL or OCSP):

- Modify the certificate cache refresh to improve performance

  The certificate cache refresh period indicates how often the certificate data store updates the certificate data in the policy store. Certificate data is cached in memory to improve CA SiteMinder® performance. Refresh the information in memory so that the data is current.

- Modify the default revocation grace period

  The default revocation grace period is the delay from when a certificate is revoked and the time the certificate becomes invalid. During the grace period, the system can use a revoked certificate before it becomes invalid. After the certificate becomes invalid, it is no longer active and CA SiteMinder® Federation Standalone cannot use it.

  If you do not specify a value for the CRL or OCSP responder grace period when adding these components, CA SiteMinder® Federation Standalone uses the default grace period. The individual grace period settings for a CRL or OCSP take precedence over this default grace period value.

Follow these steps:

1. Log in to the Administrative UI.
2. Select the Certs and Keys tab.
3. Select CDS Settings.

   The Certificate Settings dialog displays.
4. You can modify the following settings:
   - Certificate Cache Refresh Period
   - Revocation Grace Period
   - LDAP Access Timeout

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

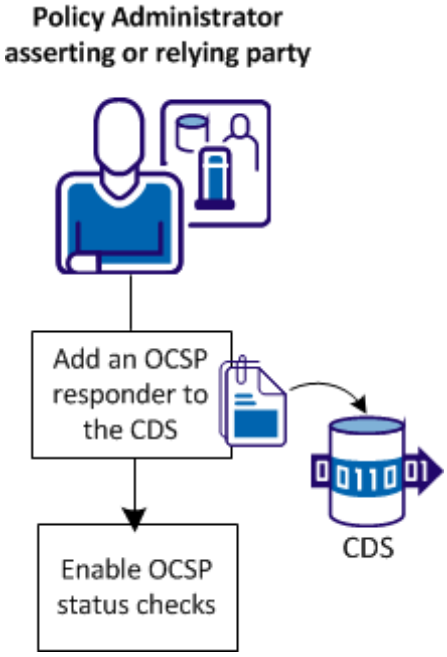
How to Verify that Certificates are Valid using OCSP

Specific federation tasks require validation for certificates in the certificate data store. These tasks include protecting the HTTP-Artifact back channel, verifying SAML messages, and encrypting SAML messages.

To check the validity of certificates, the certificate data store can use an OCSP service. OCSP uses an HTTP service that is provided by a Certificate Authority (CA) to supply the certificate revocation status on demand.

By default, CA SiteMinder® Federation Standalone does not check the revocation status of a certificate in the certificate data store. To check the revocation status through an OCSP responder, enable OCSP through the Administrative UI. When enabled, the OCSP service checks the revocation status for configured OCSP responders every 5 minutes. This default frequency is configurable.

The following figure shows the OCSP configuration steps:



The configuration process is as follows:

1. Add an OCSP responder to the CDS. (see page 130)
2. Enable OCSP status checking. (see page 131)

OCSP Prerequisites

Set up the following components to use OCSP for certificate validation:

- Set up an OCSP responder.
- Store an OCSP trusted responder certificate in the certificate data store. The responder certificate validates the signature of an OCSP response returned to CA SiteMinder® Federation Standalone. This certificate is a single trusted verification certificate or a collection of certificates.

  Obtain these certificates from your CA in a communication that is separate from an OCSP transaction.

  CA SiteMinder® Federation Standalone can work with any OCSP response that is signed using SHA-1 and the SHA-2 family of algorithms (SHA224, SHA256, SHA384, SHA512).

  The OCSP responder can include the signature verification certificate with the response. CA SiteMinder® Federation Standalone then validates the certificate and the response signature with the trusted certificate in the certificate data store.

  If a signature verification certificate is not in the response, CA SiteMinder® Federation Standalone verifies the signature with the certificate or collection of certificates in the certificate data store.

  You configure OCSP in the Administrative UI and are required to specify the location of the certificate or the collection of certificates.
- Store the CA certificate that issued the user certificate in certificate data store. This CA certificate validates the user certificate.
- (Optional) Store the private key/certificate pair that CA SiteMinder® Federation Standalone uses to sign the OCSP request is in the certificate data store.

Add an OCSP Responder to the CDS

Add an OCSP responder record to the certificate data store for each responder with which CA SiteMinder® Federation Standalone interacts.

Follow these steps:

1. Log in to the Administrative UI.
2. Navigate to the Certs and Keys tab.
3. Select the OCSP Configuration option.

   The OCSP Configuration List displays.

4. Click Add.
5. Complete the fields to add an OCSP responder configuration.

   Note: Click Help for a description of fields, controls, and their respective requirements.
6. Click Save.
7. Repeat this process for each OCSP responders you want to configure.

An OCSP responder record is now in the certificate data store.

Enable OCSP Status Checks

Add an OCSP responder record to the certificate data store for each responder with which CA SiteMinder® Federation Standalone interacts.

Follow these steps:

1. Log in to the Administrative UI.
2. Navigate to the Certs and Keys tab.
3. Select the CDS Settings option.

   The CDS Configuration List displays.
4. In the OCSP Updater section, select Enabled for the OCSP Updater State field.
5. Click Save.

OCSP status checks are enabled.

Manage Certificate Cache Refresh and Grace Period

You can complete two other tasks to manage certificate validity checking (CRL or OCSP):

- Modify the certificate cache refresh to improve performance

  The certificate cache refresh period indicates how often the certificate data store updates the certificate data in the policy store. Certificate data is cached in memory to improve CA SiteMinder® performance. Refresh the information in memory so that the data is current.

- Modify the default revocation grace period

  The default revocation grace period is the delay from when a certificate is revoked and the time the certificate becomes invalid. During the grace period, the system can use a revoked certificate before it becomes invalid. After the certificate becomes invalid, it is no longer active and CA SiteMinder® Federation Standalone cannot use it.

  If you do not specify a value for the CRL or OCSP responder grace period when adding these components, CA SiteMinder® Federation Standalone uses the default grace period. The individual grace period settings for a CRL or OCSP take precedence over this default grace period value.

Follow these steps:

1. Log in to the Administrative UI.
2. Select the Certs and Keys tab.
3. Select CDS Settings.

   The Certificate Settings dialog displays.
4. You can modify the following settings:
   - Certificate Cache Refresh Period
   - Revocation Grace Period
   - LDAP Access Timeout

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

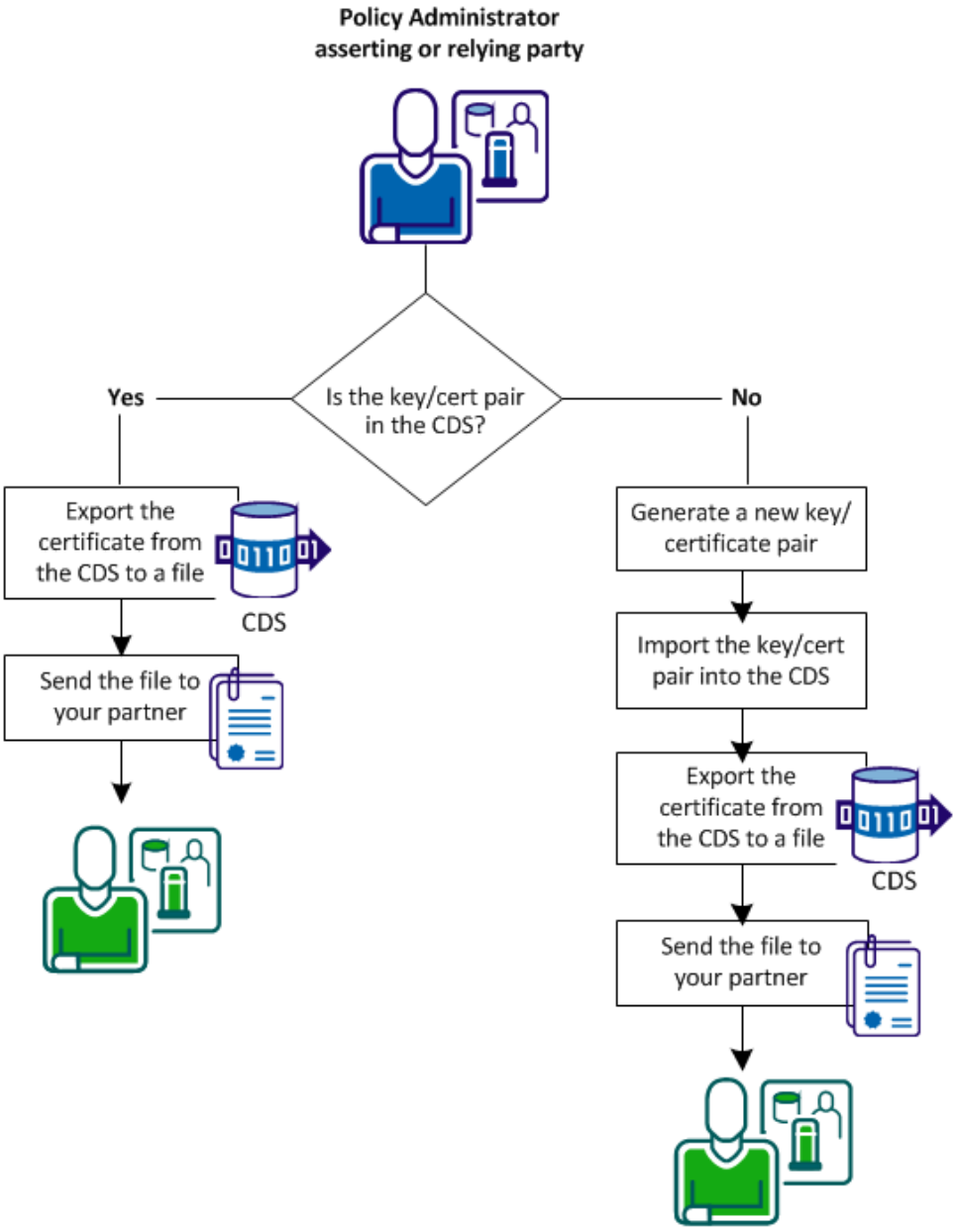
How to Send Certificates to Your Partner

The partner that signs a message has to send the associated certificate (public key) to the other partner so that partner can verify the message.

The partner that encrypts a message has to receive the certificate (public key) to from the partner expected to decrypt the message.

The procedure for sending the required certificate file to a partner depends on whether the key/certificate pair is already in the CDS.

The following figure shows the steps for sharing certificate files.



Follow these steps:

1. Generate a new key/certificate pair.
2. Import the key/certificate pair into the CDS (see page 135).
3. Export the certificate from the CDS to a file (see page 137).
4. Send the certificate file to your partner. (see page 138)

Generate a New Key/Certificate Pair Using the UI or a Third-party Tool

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

Generate a certificate request using the Administrative UI or using a third-party tool.

When you create a request using the Administrative UI, CA SiteMinder® Federation Standalone generates a private key and a self–signed certificate pair. CA SiteMinder® Federation Standalone stores this pair in the certificate data store. Using the generated request, contact a Certificate Authority and fill out the CA certificate request form, pasting the contents of the generated request into the form.

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

Follow these steps:

1. Log in to the Administrative UI.
2. From the Certs & Keys tab, select Certificate and Private Keys.
3. Click Request Certificate.
4. Complete the required fields.

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

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

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

Import the Key/Cert Pair into the CDS

The procedure for importing the key/certificate pair varies. Refer to the appropriate procedure:

- Import a key/cert pair from an existing file.

  You have a local file on your system.

- Import a signed certification response (see page 123).

  If you generated a key/certificate pair from the Administrative UI, import the certificate from the signed response sent by the Certificate Authority.

Import a Key/Certificate Pair from an Existing File

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

CA SiteMinder® Federation Standalone treats a certificate that you import as a trusted certificate. The exceptions are self-signed certificates:

- If the system identifies a V3 self-signed certificate as a CA certificate, the certificate is treated as a CA certificate. This behavior occurs even though you initiate the import from the Certificate/Private Key dialog.
- CA SiteMinder® Federation Standalone treats the certificate as a trusted certificate:
  - If CA SiteMinder® Federation Standalone does not identify a V3 self–signed certificate as a CA.
  - If the certificate is a V1 self–signed certificate.

Follow these steps:

1. Log in to the Administrative UI.
2. From the Certs & Keys tab, select Certificates and Private Keys.

   The View Certificates and Private Keys dialog opens.

3. Click Import New and follow the wizard.

   Note: You can click Help for a description of fields, controls, and their respective requirements.

   Be aware of the following items as you complete the wizard:

   - You can import a single file with a key and certificate in it or separate key and certificate files. Select the appropriate option button for the file you are using.
   - To import a self-signed certificate as a Certificate Authority certificate, set the Use as CA option button to Yes. The certificate is imported as a CA certificate and is not available for when configuring partnerships (for example, for signing or encryption).

     Otherwise, accept the default No setting to import the certificate as a trusted certificate that is available when configuring partnerships.
   - For a trusted certificate file in DER (binary) format, the file can contain one or more certificate entries. For a trusted certificate file in PEM (base 64) format, CA SiteMinder® Federation Standalone expects one certificate per file.

     The standard extension for a file in DER or PEM format is *.crt or *.cer.
   - If you are using a .p12 file, you are required to fill in a password. CA SiteMinder® Federation Standalone processes a .p12 or .pfx file as a file containing key/certificate pairs.
   - For each entry you plan to add to the certificate data store, enter the alias you want to associate with that entry. If you select multiple entries, each requires a unique alias.

4. At the Confirm step, review the information and click Finish.

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

Import a Signed Certificate Response

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

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

Follow these steps:

1. From the Certs & Keys tab, select Certificate and Private Keys.

   The View Certificates and Private Keys dialog opens.

2. Search for the self-signed entry with the same alias.

3. Select Action, Update Certificate next to the entry that contains the self-signed certificate.

   The wizard for importing certificates and keys displays.

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

4. Browse to the file you want. You can use a:

   - .p7 or .p7b file that contains the signed certificate and the corresponding certificate chain.
   - .cer or .crt file (base64 PEM file) with the signed certificate without the certificate chain.

5. Select the appropriate entry.
6. At the Confirm step, review the certificate information and click Finish.

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

Export Certificates from the CDS using the Administrative UI

You can export a private key/certificate pair to a file and send the certificate file (public key) to your federation partner. The partner can use the certificate to verify the signature of assertion responses created with the associated private key or encrypt a response to be decrypted with the associated private key.

Important! If you export the private key as part of a backup, never share it with anyone else.

Follow these steps:

1. Log in to the Administrative UI.
2. From the Certs & Keys tab, select Certificates and Private keys.

   The View Certificates and Private Key window displays.

3. Select Action, Export for the entry in the Certificate and Private Key List you want to export.

   The Export Key Store Entry dialog displays.

4. Select the format of the file you want to create from the exported data.

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

5. Select the file format.
6. Click Export.

   You are prompted to open or save the file on the local system.

   CA SiteMinder® Federation Standalone generates the encoded file content representing the key or certificate.
7. Send the file to your partner (see page 138).

Send the Certificate File to your Partner

After exporting the encoded file with the certificate, send this file to your federation partner. Your partner has to import this certificate to handle verification or encryption of federation messages.

Update Certificates in the Certificate Data Store

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

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

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

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

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

Certificate Authority (CA) Certificate Usage

The federation system uses Certificate Authority certificates to verify the following items:

- Whether an SSL server certificate for an SSL connection that secures the SAML HTTP-Artifact back channel is trusted.
- For HTTP-Artifact single sign-on, secure the back channel with an SSL connection. The embedded web server for the federation system can verify that the SSL connection is secured by a trusted certificate by validating the certificate of the Certificate Authority. This certificate must be stored in the certificate data store.
- Whether a certificate revocation list is valid.

  CRLs are acquired from a Certificate Authority. The certificate of the corresponding CA is required to validate the CRL before it can be trusted. The CRL is stored in the data store for use at runtime.

A default set of common root and intermediate CA certificates are shipped with the product for these purposes.

Import a CA Certificate

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

Any certificate that you import is treated as a CA certificate. The exceptions are self-signed certificates:

- If the system identifies a V3 self-signed certificate as a non-CA certificate, the certificate is treated as a trusted certificate. This behavior occurs even though you initiated the import from the Import CA Certificate dialog.
- If the system identifies a V1 self-signed certificate, the certificate is treated as a CA certificate.

Note: If you are importing a root CA certificate, import all root CA certificates in the chain if they are part of a trust chain.

To import a CA certificate

1. Log in to the Administrative UI.
2. Select Certs & Keys, Authorities.

   The Certificate Authorities List displays.
3. Click Import New.

   The Import CA Certificate dialog displays.

   Note: Click Help for a description of fields, controls, and their respective requirements.
4. Follow the wizard to import a new entry.
5. 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.

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

Troubleshoot Certificate Signature Verification for Back Channel Communication

Symptom:

HTTP-Artifact is the profile in use for single sign-on. The asserting party is communicating to the relying party over an SSL back channel. The relying party must verify the signature of the server certificate at the asserting party to communicate over the SSL back channel.

The following error is logged for a failure to verify the signature of the server certificate:

```
[Dispatcher object thrown unknown exception while processing the
request message. Message: Certificate not verified..]
```

Solution:

The relying party must import the root CA certificate into the certificate data store. This certificate is required to verify the signature of the server certificate at the asserting party. For verification, import the root CA that signed the server certificate.

Verify the following information about the CA certificate that is imported for verification:

- Does the root CA certificate have the same issuer and subject DN? If not, the certificate is an intermediary root CA. Import all root CA certificates in the trusted chain.
- Check that the issuer of the asserting party server certificate matches the subject and issuer of the root CA you have imported.

Chapter 8: Partnership Creation and Activation

This section contains the following topics:

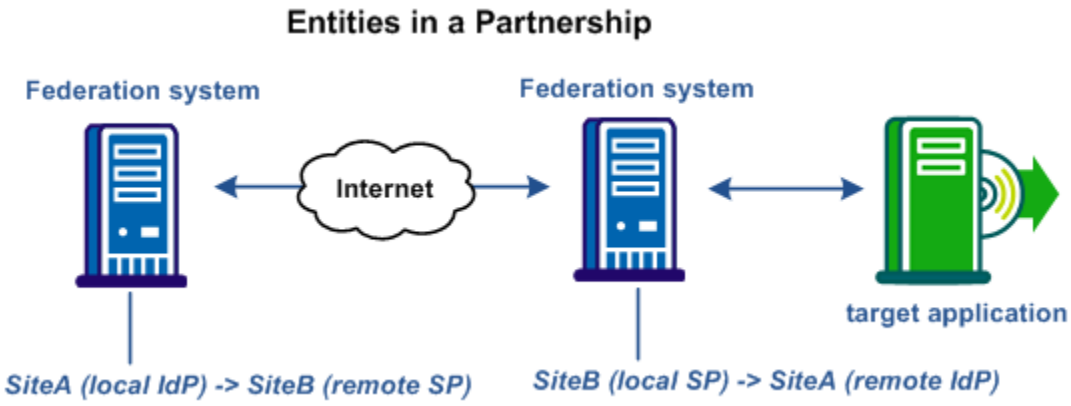
Partnership Creation (see page 143)
Partnership Definition (see page 144)
Partnership Identification and Configuration (see page 145)
Partnership Confirmation (see page 146)
Partnership Activation (see page 147)
Exporting a Partnership (see page 147)

Partnership Creation

The main purpose of CA SiteMinder® Federation Standalone is to establish a partnership between two organizations so they can share user identity information and can facilitate single sign-on (SSO). A partnership consists of two entities at different sites—one local and one remote. Either entity can assume the role of the asserting party, the side which produces assertions or the relying party, the side which consumes assertions.

If CA SiteMinder® Federation Standalone is installed at both sites, each site must define a partnership. For each local asserting party-to-relying party partnership at one site, there has to be a reciprocal local relying party-to-asserting party partnership at the partner site. The two definitions define a single partnership.

In the following figure, SiteA has been configured as the local SAML 2.0 IdP and has specified SiteB as the remote SAML 2.0 SP. SiteB has been configured as the local SAML 2.0 SP, and SiteA is its remote SAML 2.0 IdP.



Note: An asserting party can have partnerships with more than one relying party and a relying party can establish partnerships with more than one asserting party.

Creating a federation partnership consists of the following steps:

1. Specify the partnership type.
2. Configure the following partnership details:
    a. Partnership name and the participating entities
    b. Federation users (local asserting party only)
    c. Name ID format and other assertion attributes (local asserting party only)
    d. User identification (local relying party only)
    e. Single sign-on (SSO)
    f. Single logout (SLO) – SAML 2.0 only
    g. Signing
    h. Encryption – SAML 2.0 only

Partnership Definition

The federation partnership definition specifies which federation role is local, and which federation role is remote.

Follow these steps:

1. Log in to the Administrative UI.
2. From the Federation tab, select Partnerships.
3. Click Create Partnership in the Federation Partnership List section.

    Note: Click Help for a description of fields, controls, and their respective requirements.
4. Select one of the following options:
    - SAML2 IDP->SP (Identity Provider is local).
    - SAML2 SP->IDP (Service Provider is local).
    - SAML1.1 Producer ->Consumer (Producer is local).
    - SAML1.1 Consumer ->Producer (Consumer is local).
    - WSFED IP->RP (Identity Provider is local).
    - WSFED RP->IP (Resource Partner is local).

The Create Federation Partnership dialog displays the first step in partnership configuration.

Partnership Identification and Configuration

In the Configure Partnership step of the wizard, identify the partnership by naming the partnership and specifying the local and remote entities.

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

Follow these steps:

1. Enter a name for the partnership. You can use alphanumeric characters, underscores, hyphens, and periods in the name. Spaces are not allowed.
2. (Optional) Type a description.
3. Select a local entity from the local list if you have already configured an entity. If not, click Create Local Entity.
4. Select a remote entity from the remote list if you have already configured an entity. If not, click Create Remote Entity.

   Note: This step can be deferred if you are planning to create the remote entity by importing metadata later.
5. (Optional) Enter the Skew Time in seconds.

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

   The system uses the skew time and the SSO validity duration to determine how long an assertion is valid.
6. Select one or more user directories from the Available Directories list and move them to the Selected Directories list.

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

   Important! To use an ODBC database as a user directory, define an SQL Query scheme and valid SQL queries. These steps are necessary before you can select it as a user directory.
7. Click Next to continue through the partnership wizard. The steps of the wizard let you configure various features of a partnership, some features are required, and some are optional. The configuration details for these features are described in subsequent sections of this guide.

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

Editing Entities from the Partnership

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

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

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

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

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

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

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

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

Partnership Confirmation

Review the partnership configuration before saving it.

Follow these steps:

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

The partnership configuration is complete.

Partnership Activation

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

Follow these steps:

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

   Note: Activate is only available for a partnership in DEFINED or INACTIVE status, and Deactivate is only available for a partnership in ACTIVE status.
3. Click Yes to confirm your selection.

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

Important! Deactivate a partnership before you modify it.

Exporting a Partnership

You can use metadata as a basis for creating remote entities and forming a partnership. Metadata makes the partnership configuration more efficient because many aspects of an entity are already defined in the metadata file. The file can then be imported to create a new partnership or remote entity.

You do not have to complete a partnership before exporting it. You can configure a portion of the partnership and then export it.

In the Administrative UI, you can export metadata from an existing partnership entry.

Note: In the Administrative UI, you can export metadata from an existing local asserting or relying entity. When you export SAML 1.1 data, the terms that are used in the resulting metadata file are SAML 2.0 terms. This convention is part of the SAML specification. When you import the SAML 1.1 data, the terms are imported correctly using SAML 1.1 terminology.

When exporting from the partnership, the selected partnership is used as the basis of the export. You are not allowed to define a new partnership name. The system uses the name from the selected partnership.

Follow these steps:

1. From the Federation tab, select Partnerships.
2. Click the Action pull-down menu next to the appropriate entry in the list and select Export Metadata.

3. Complete the fields on the dialog.

   If you are exporting a partnership in ACTIVE status, most of the fields are read-only; only the Validity Duration field and the alias drop-down list can be edited.

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

4. Click Export to finish.
5. A dialog prompting you to open or save the metadata file displays. You can open it to view it.
6. Save the data to an XML file on your local system.

The metadata is exported to the specified XML file.

Chapter 9: Federated User Identification for a Partnership

This section contains the following topics:

Federation Users Configuration at the Asserting Party (see page 149)
User Identification at the Relying Party (see page 153)

Federation Users Configuration at the Asserting Party

The Federation Users dialog is the second step in the partnership wizard when the local entity is the asserting party. This step lets you specify which users are authorized to access target resources at the remote site. Additionally, you can enable the CA SiteMinder® Connector for CA SiteMinder® Federation Standalone integration with CA SiteMinder®.

The CA SiteMinder® Connector is a software component that enables a deployed CA SiteMinder® system to integrate with CA SiteMinder® Federation Standalone. If CA SiteMinder® Federation Standalone is at the asserting party, the CA SiteMinder® Connector can create a CA SiteMinder® Federation Standalone session from a CA SiteMinder® session. To establish the CA SiteMinder® session, CA SiteMinder® authenticates the user first and then the user visits the asserting party.

You can enable the CA SiteMinder® Connector on a per-partnership basis; however, only one global connector configuration applies to all partnerships. The connector is available only when the check box in the Deployment Settings is selected and a configuration is defined. You access the Deployment Settings from the Infrastructure tab in the UI. After enabling the Connector globally, CA SiteMinder® Federation Standalone evaluates the partnership configuration to determine whether the Connector is enabled. The partnership uses the global Connector configuration.

To disable the Connector for the partnership, clear the check box at the partnership level. To disable the Connector globally, disable it in the Deployment Settings.

Important! If the Connector is disabled at the global level, CA SiteMinder® Federation Standalone ignores the check box at the partnership level.

Configure Federation Users

Federation users are those users that are allowed to access protected federated resources.

Follow these steps:

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

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

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

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

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

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

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

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

6. (Optional) Configure the CA SiteMinder® Connector settings:

   a. If CA SiteMinder® Federation Standalone is integrating with an existing SiteMinder deployment, enable the CA SiteMinder® Connector by selecting the check box.

b. (Optional) Clear the Enforce UserDN and Directory Name Comparison so that the CA SiteMinder® Federation Standalone or CA SiteMinder® uses a Universal ID to retrieve a user record. The Universal ID enables the user directories to be physically different and of different types. Use of the Universal ID is sufficient to regard the retrieved user record as the correct record.

Note: If you rely on the Universal ID, each user must have a unique Universal ID. If the Universal IDs are not unique, the system accessing the user record can retrieve the wrong record.

If you leave the check box selected (the default), CA SiteMinder® Federation Standalone and CA SiteMinder® must use the same physical directory. The name for both of these directories must be the same for user store lookups. The entity authenticating the user compares the information that the user provides against the UserDN and the Directory Name of the user record.

The selection of users is complete.

7. Click Next.

The Assertion Configuration dialog displays.

Examples of User Class Entries

LDAP Examples

Use the LDAP filter syntax when specifying entries.

User Class	Valid Entry
User	Distinguished name of a user. Example: uid=user1,ou=People,dc=example,dc=com
Group	Group chosen from the list. Example: ou=Sales,dc=example,dc=com
Organization Unit	Organizational unit chosen from the list. Example: ou=People,dc=example,dc=com
Filter User Property	LDAP filter. The current user is the starting point for the search. **Example 1:** mail=user@example.com **Example 2:** (|(mail=*@.example.com)(memberOf=cn=Employees,ou=Groups,dc=example,dc=com))

User Class	Valid Entry
Filter Group Property	LDAP filter. The current user gets authorized if they are a member of one of the groups matching the filter. The objectclasses for groups as configured in the SiteMinder registry are combined with the filter. **Example 1:** To authorize users that are members of a group with a business category of "CA Support", enter: businessCategory=CA Support **Example 2:** To authorize users that are members of a group with a description containing "Administrator" and a business category of "Administration", enter: (|(description=*Administrator*)(businessCategory=Administration)) **Note:** Not all attributes of a group work as a search criterion.
Filter OU Property	LDAP filter. The current user gets authorized if they belong to an organizational unit that matches the filter. The objectclasses for organizational units as configured in the SiteMinder registry are combined with the filter. **Example 1:** To authorize users within an organizational unit with a postal code of "12345", enter: postalCode=12345 **Example 2:** To authorize users in an organizational unit with a preferred delivery method ending with "phone" and a locality of "London", enter: (|(preferredDeliveryMethod=*phone)(l=London))
Filter Any	LDAP filter. The current user gets authorized if they match the filter. **Example 1:** To authorize users with a department of "CA Support", enter: department=CA Support **Example 2:** To authorize users who are members of the group "Administrators" and have a department number of "123" or "789", enter: (&(memberof=cn=Administrators,ou=Groups,dc=example,dc=com)(|(departmentNumber=123)(departmentNumber=789)))

ODBC Examples

Use the SQL syntax when specifying queries.

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

User Identification at the Relying Party

At the relying party, the partner must be able to locate a user in the local user directory. Locating the user in the user directory is the process of disambiguation. Configure the identity attribute for user disambiguation in the User Identification dialog.

CA SiteMinder® Federation Standalone can employ one of the following methods for the disambiguation process:

- Extract the Name ID value from the assertion.
- Use the value of a specific attribute from the assertion.
- Use the value obtained by an Xpath query.

  The Xpath query locates and extracts an attribute other than the Name ID from the assertion.

After you determine which attribute is extracted from the assertion, include this attribute in a search specification, which CA SiteMinder® Federation Standalone uses to locate a user in the user store. After a successful disambiguation process, CA SiteMinder® Federation Standalone generates a session for the user.

For SAML 2.0, you can also configure the AllowCreate feature (see page 156), which lets an asserting party create a user identifier.

Single sign-on can be initiated by the relying party sending an authentication request (AuthnRequest) to the asserting party. In this request, the relying party can ask that the asserting party include a particular user attribute in the assertion. However, the value of the required attribute may not be available in the asserting party user record.

If the authentication request from the relying party includes the Allow/Create attribute and the asserting party is configured to create a new identifier, the asserting party generates a unique value as the NameID. This value is placed in the assertion and sent back to the relying party.

In the User Identification dialog, you can also enable the CA SiteMinder® Connector.

The CA SiteMinder® Connector is a software component included with CA SiteMinder® Federation Standalone. It enables a deployed CA SiteMinder® system to integrate with CA SiteMinder® Federation Standalone. If you integrate CA SiteMinder® and CA SiteMinder® Federation Standalone at a relying party, CA SiteMinder® does not rechallenge users authenticated by CA SiteMinder® Federation Standalone when they request CA SiteMinder®-protected resources. There is no authentication rechallenge because the Connector and a custom CA SiteMinder® authentication scheme at the Policy Server enable the creation of a CA SiteMinder® session for users authenticated by CA SiteMinder® Federation Standalone.

You can enable the CA SiteMinder® Connector on a per-partnership basis; however, only one global SiteMinder Connector configuration applies to all partnerships. The Connector is available only when the check box in the Deployment Settings is selected and a configuration is defined. You access the Deployment Settings from the Infrastructure tab in the UI. After enabling the Connector globally, CA SiteMinder® Federation Standalone evaluates the partnership configuration to determine whether the connector is enabled. The partnership uses the global Connector configuration.

To disable the Connector for the partnership, clear the check box at the partnership level. To disable the Connector globally, disable it in the Deployment Settings.

Important! If the Connector is disabled at the global level, CA SiteMinder® Federation Standalone ignores the check box at the partnership level.

Configure User Identification at the Relying Party

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

Follow these steps:

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

1. Select one of the following attributes:
   - Name ID
   - An attribute from a previously populated drop-down list

     If the remote asserting entity was created based on metadata that contained attributes, the list is populated.
   - An attribute you enter

     This option is most likely used when metadata is not available and the remote asserting entity does not include any attributes.
   - An Xpath
2. (Optional—SAML 2.0 only) Select Allow IDP to create user identifier.

   This attribute instructs the asserting party to generate a new value for the NameID, if this feature is enabled at the asserting party. The Name ID format configured at the asserting party must be a persistent identifier. This new value for the NameID is included in the assertion that the asserting party returns to the relying party.
3. Specify an LDAP or ODBC search specification. If both directories are present, configure search specifications for both.

   LDAP Example

   ou=%s,o-ca

   ODBC Example

   name=%s

   In the ODBC search specification field, the value from the user store that replaces the %s in the search string can contain an equals sign (=). If the value contains an equals sign, prepend the value **user=** at the beginning of the entry. For example, if the value for ElectronicMail in the user store is CN=catechnologies, enter **user=ElectronicMail=%s** in the ODBC search specification field. The addition of user= enables the policy engine to interpret the string properly.
4. (Optional) Configure the CA SiteMinder® Connector settings:
   a. If CA SiteMinder® Federation Standalone is integrating with an existing SiteMinder deployment, enable the CA SiteMinder® Connector by selecting the check box.

b. (Optional) Clear the Enforce UserDN and Directory Name Comparison so that the CA SiteMinder® Federation Standalone or CA SiteMinder® uses a Universal ID to retrieve a user record. The Universal ID enables the user directories to be physically different and of different types. Use of the Universal ID is sufficient to regard the retrieved user record as the correct record.

Note: If you rely on the Universal ID, each user must have a unique Universal ID. If the Universal IDs are not unique, the system accessing the user record can retrieve the wrong record.

If you leave the check box selected (the default), CA SiteMinder® Federation Standalone and CA SiteMinder® must use the same physical directory. The name for both of these directories must be the same for user store lookups. The entity authenticating the user compares the information that the user provides against the UserDN and the Directory Name of the user record.

5. Click Next to continue with partnership configuration.

Employ AllowCreate for User Identification (SAML 2.0)

The SAML 2.0 AllowCreate feature is an optional setting in the User Identification configuration at the SP. Including an AllowCreate attribute in an authentication request lets an Identity Provider create a user identifier for the SP.

An SP can initiate single sign-on by sending an authentication request to the Identity Provider. As part of the request, a Service Provider can include 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 unique persistent identifier for the NameID. Enable the Allow/Create feature at the Identity Provider for it to generate the identifier. The Identity Provider returns the assertion with the unique identifier back to the SP.

You can enable an AllowCreate query parameter to supersede the value of the AllowCreate attribute. Use of a query parameter lets you override the configured AllowCreate setting without deactivating, editing, and reactivating the partnership. The query parameter makes the implementation of the feature more flexible.

Chapter 10: Assertion Configuration at the Asserting Party

This section contains the following topics:

Assertion Configuration (see page 157)
Configure Assertion Options (see page 158)
Assertion Attribute Configuration Examples (see page 160)
How To Add Session Attributes to an Assertion (see page 160)
How to Configure Claims Transformation at the Asserting Party (see page 163)
Customize Assertion Content (see page 173)

Assertion Configuration

The Assertion Configuration step of the partnership wizard defines the configuration of the following settings:

Name ID

The Name ID attribute, a required assertion attribute, identifies a user in a unique way. The Name ID format indicates the identifier type that the federated partners support. The Name ID type specifies the user profile attribute that is associated with the name ID format. The user profile attributes come from a user store or the session store.

Assertion Attributes

Servlets, web applications, or other custom applications can use attributes to display customized content or enable other custom features. When used with web applications, attributes can limit the activities of a user at the relying party. For example, an attribute variable named Authorized Amount is set it to a maximum dollar amount that the user can spend at the relying party.

Attributes are designated in an <AttributeStatement> element or an <EncryptedAttribute> element. Attributes take the form of name/value pairs. Attributes can also be made available as HTTP headers or HTTP cookies.

Note: Attributes statements are not required in an assertion.

You can configure different types of attributes for an attribute statement. The types of attributes include:

- user attributes
- DN attributes
- static data
- session attributes

  Session attributes are available for assertions only if they are persisted in the session store.

You can also configure an expression to transform assertion attributes. This capability is called claims transformation.

When the relying party receives the assertion, it makes the attribute values available to applications.

Assertion Generator Plug-in

Typically, attributes come from user directory records, but an assertion can contain attributes from other sources, such as an external database or application content. You can write an assertion generator plug-in that pulls in attributes from various sources. The assertion generator plug-in is a piece of custom code that you write according to the Assertion Generator Plug-in interface.

For information about writing a plug-in, see the *Programming Guide for the Federation Java SDK*.

Configure Assertion Options

Configure assertion options at the asserting party.

Follow these steps:

1. Navigate to the Assertion Configuration step of the partnership wizard.
2. Configure the settings in the Name ID section.

   The relying party uses these values to interpret the Name ID value in the assertion.

   Depending on the selected NameID Type option, complete the entry with a proper value.

   Static attribute

   Enter any constant string in the Value field.

   User Attribute

   Enter a valid user store attribute in the Value field. For example, mail.

Session attribute

Enter a valid session store attribute in the Value field.

DN Attribute (LDAP only)

Enter a valid LDAP user directory attribute in the Value field. Also, enter a valid DN in the DN specification fields. For example, the DN attribute is cn=JaneDoe and the specification is ou=Engineering,o=example.com.

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

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

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

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

   For help filling out the table, view some assertion attribute examples. Click Help for detailed information about each column in the attribute table.

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

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

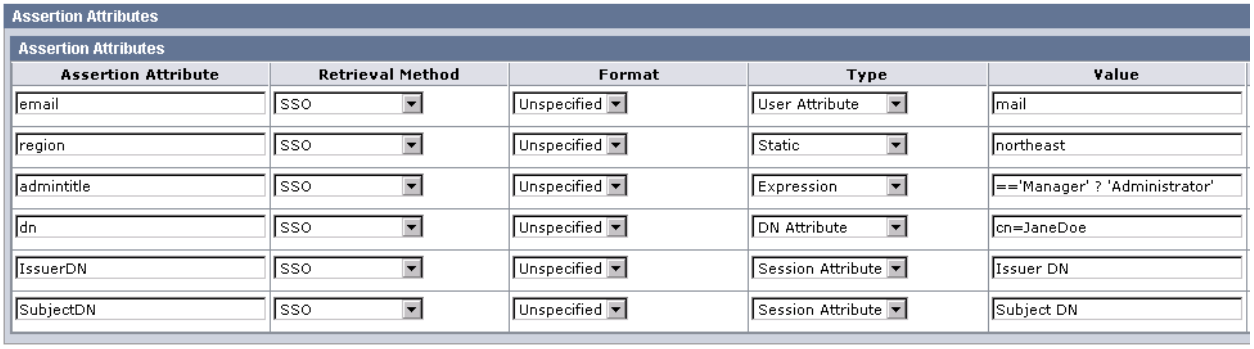
   To write a plug-in, see the *Programming Guide for the Federation Java SDK*.

6. Click Next to continue with the partnership configuration.

Assertion Attribute Configuration Examples

The following graphic shows some examples of assertion attribute entries. This screen is for a SAML 2.0 partnership. The SAML 1.1 screen is similar, but the Retrieval Method and Format columns are missing. A Namespace column exists instead.

Note: The DN Attribute example includes a DN Specification column, with the entry ou=Engineering,o=example.com. This column is not visible in this graphic.



How To Add Session Attributes to an Assertion

The Policy Server uses the session store to persist dynamic user information after a user is authenticated. The stored information includes authentication context information, SAML attributes, third-party IdPs that authenticate users, and claims from an OAuth authentication. The Policy Server can use this information for generating user tokens or making policy decisions.

For federated single sign-on, the Policy Server can add the attributes from the session store to an assertion to customize the requested application.

Session attributes are stored for the following deployments:

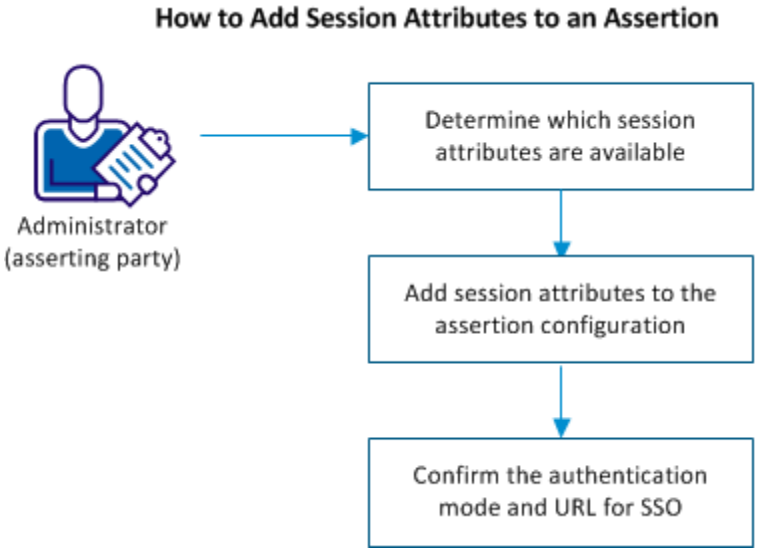
- Non-delegated authentication deployments.

  A local system or an external third party authenticates users, but the system regards it as a local authentication. Local authentication deployments require that the authentication mode is local in the single sign-on configuration. Also, an access policy must protect the authentication URL. The authentication scheme in the policy is configured to persist session attributes.

- Delegated authentication deployments

  An external third party can authenticate a user. The third-party partner returns user information, which gets stored in the session store.

The following figure shows the steps that are required to configure session attributes and add then to assertions.



Complete the following steps for session attribute support:

1. Determine which session attributes are available. (see page 161)
2. Add session attributes to the assertion configuration. (see page 161)
3. Confirm the authentication mode and URL for SSO.

Determine which Session Attributes are Available

As the federation administrator, identify the session attributes used by the partnership. Work with the authentication source, such as a database or user directory so you are familiar with the available attributes.

Add Session Attributes to the Assertion Configuration

Add session attributes to the assertion configuration. The configuration is at the asserting party, such as the IdP-to-SP partnership.

Follow these steps:

1. Log in to the Administrative UI.
2. Navigate to the Assertion Configuration step of the partnership wizard.
3. In the Assertion Attributes section, click Add Row.

4. To configure a session attribute, complete the settings in the table. For example:

   Assertion Attribute

   IssuerID

   Retrieval Method

   SSO

   Format

   Unspecified

   Type

   Session Attribute

   Value

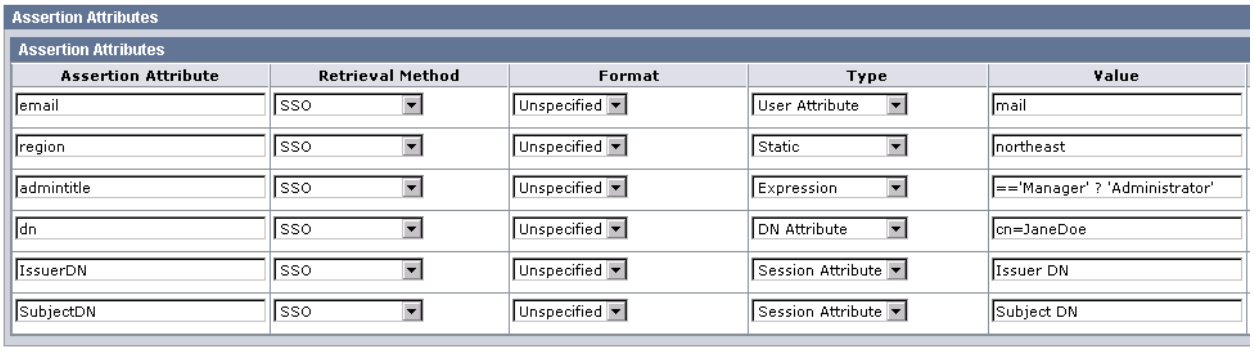
   IssuerID

   Click Help for detailed information about the attribute table.

5. Add rows for as many entries as needed.
6. (Optional). Select Encrypt to encrypt the attribute.
7. Click Next to move to the SSO and SLO step.

Session Attribute Examples in the Administrative UI

The last two entries of the following graphic show examples of session attribute entries. This screen is for a SAML 2.0 partnership. The SAML 1.1 screen is similar, but the Retrieval Method and Format columns are missing. A Namespace column exists instead.



Confirm the Authentication Mode and URL for SSO

Confirm that the partnership has the authentication mode and authentication URL set correctly.

Note: This procedure assumes that the other necessary SSO settings are configured.

Follow these steps:

1. Navigate to the SSO and SLO step of the partnership wizard.
2. In the Authentication section, verify the settings of the following fields:

   Authentication Mode

   Local

   Authentication URL

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

   http://*myserver*.idpA.com/siteminderagent/redirectjsp/redirect.jsp

   myserver

   Identifies the web server with the Web Agent Option Pack or the SPS federation gateway. The redirect.jsp file is included with the Web Agent Option Pack or SPS federation gateway that is installed at the asserting party.
3. Navigate to the Confirm step and click Finish.

How to Configure Claims Transformation at the Asserting Party

Claims transformation manipulates claims during a federated single sign-on transaction. Claims, also known as attributes, help customize the attributes and improve the user experience at a partner.

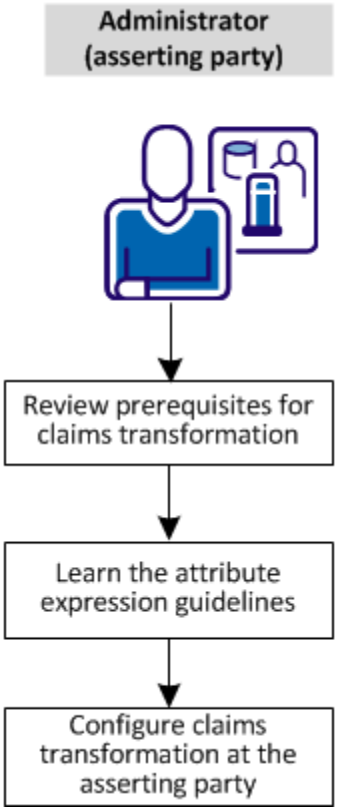
Modifying assertion attributes lets the relying party adapt user information so a target application can use it. For example, claims transformation can associate roles at different partners in different domains. In one domain, a user is an engineering manager and belongs to a group named EngineerAdmins. However, the relying party identifies the same role as DevelAdmins. The asserting party alters the role attribute before issuing the assertion. The user is now identified with the DevelAdmins role, which the relying party application can understand.

Claims transformation occurs at the local asserting party during the assertion generation process. You configure the feature on a per-partnership basis. An assertion can be modified whether a local or remote party generates the assertion. Claims are transformed based on an expression that you configure for the partnership. The expression relies on user information from the user store and the CA SiteMinder® session store.

The software can perform three different modifications to assertion attributes:

- **Transformation:** Changing the value of an assertion attribute to a different value.
- **Addition:** Adding an assertion attribute if it does not exist already.
- **Deletion:** Deleting an assertion attribute on a conditional basis.

The following figure shows the configuration steps:



To set up claims transformation, perform the following steps:

1. Review the prerequisites for claims transformation. (see page 165)
2. Learn the attribute expression guidelines (see page 165).
3. Configure claims transformation at the asserting party (see page 167).

Prerequisites for Claims Transformation

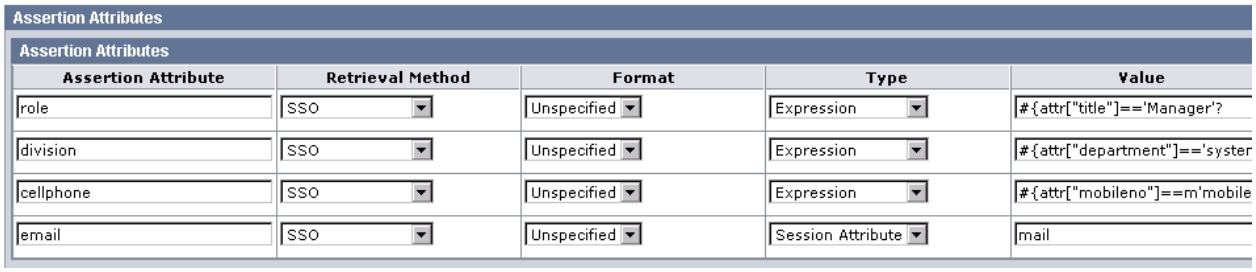
Before you configure claims transformation, consider the following prerequisites:

- Be familiar with the user store and session store attributes available.
- Determine which attributes the relying party expects to receive in an assertion.
- Be familiar with Java Unified Expression Language (JUEL), an open source version of the Unified Expression Language.

Learn the Attribute Expression Guidelines

Expressions are rules that instruct the software how to manipulate assertion attributes. The expression directs the software to modify, add, or delete assertion attributes. You construct expressions using the Java Unified Expression Language (JUEL). A JUEL expression evaluator examines the configured expressions and generates the resulting assertion attributes.

Define expressions in the Assertion Attributes table of the Administrative UI. Access this table by navigating to the Assertion Configuration step of the partnership wizard. This table is shown in the following figure:



Enter expressions in the Value column of the assertion attributes table. All attributes in an expression are user store or session store attributes.

Typically, the expression operates on a conditional basis. If the condition is met, the designated claims modification occurs. For example, an incoming assertion contains the "role" attribute. The expression to modify the "role" assertion attribute is:

#{attr["title"] == 'manager' ? 'administrator' : attr["title"]}

The first part of the expression **#{attr["title"] == 'manager'** tells the software to determine whether the logged-in user has the title "manager." The lookup is done in the user directory. If this condition is met, the second part of the expression, **? 'administrator' :** assigns the value "administrator" to the role assertion attribute. If the condition is not met, the last part of the expression, **attr["title"]}** indicates that the value of the user attribute "title" remains "manager." The value "manager" is assigned to the assertion attribute "role."

Note: You can use static values in an expression in place of the syntax attr["title"], such as 'administrator' in the previous example.

The example assumes the "role" attribute is already in the assertion. Therefore, the expression is a transformation of an existing attribute. If "role" is not part of the assertion, the software adds the role attribute to the assertion.

Expression Syntax

Construct expressions using the proper syntax:

- Represent a user store attribute with the string **attr**["*attribute_name*"].
- Represent a session store attribute with the string **session_attr**["*attribute_name*"].
- Delete a claim using the argument 'DELETE'.

Use lower-case text for the **attr** and **session_attr** prefixes. Attribute names are not case-sensitive.

Additionally, be aware of these conditional JUEL operators:

Operator	Meaning
conditional value ? value1 : value2	The conditional value resolves to either value1 or value 2.
!=	Does not equal
==	Equals

Important! The attribute in the expression must be available in the user directory or the session store. If an attribute is incorrect, the system simply includes blanks for corresponding attributes. The assertion generation does not fail.

For more examples of expressions, read the section Configure claims Transformation at the Asserting Party.

Configure Claims Transformation at the Asserting Party

Define expressions at the partnership level. The result of these expressions modifies, adds, or deletes attributes from assertions. After the rules are defined, the assertion is modified and sent to the relying party. If you do not configure claims transformation, assertion attributes are passed "as is" to the relying party.

Follow these steps:

1. Log in to the Administrative UI.
2. Click Federation, Partnerships.
3. Select a partnership that you want to modify. Eligible partnerships include:
   - Local Producer to Remote Consumer
   - Local IdP to Remote SP
   - Local IP to Remote RP
4. Navigate to the Assertion Configuration step in the partnership wizard.

   In the Assertion Attributes section, click Add Row.
5. Pay particular attention to the following fields in the row. Click Help for detailed descriptions of each field.

   Assertion Attribute

   Enter an assertion attribute. All values in this column are assertion attributes. An attribute that is already in the assertion remains in the assertion, but it is set to a new value based on the configured expression. The attribute is removed from the assertion only if you configure a DELETE expression.

   Retrieval Method

   Keep the default, SSO.

   Format

   Designates the format for the attribute being added to the assertion. The format options vary depending on the SAML profile for the entity.

   Type

   Expression

   Always use this value for claims transformation.

Value

Enter an expression that reflects how you want the assertion attribute is modified.

Review the guidelines about constructing claims expressions and the following examples:

- Transform a claim in an assertion. (see page 168)
- Add a claim to an assertion (see page 170).
- Delete a claim from an assertion (see page 171).

6. (Optional for SAML 2.0 and WSFED with token type SAML 2.0). To encrypt assertion attributes, select Encrypt. The asserting party encrypts the assertions using the certificate that is specified in the partnership configuration.

   The relying party decrypts the assertion attributes using the private key that is associated with the certificate.

7. Add as many rows as you like for the assertion attributes you want to configure.

Claims transformation is implemented based on the configured entries in the partnership.

Transform a Claim in an Assertion

Transforming a claim changes an assertion attribute value to another value.

Note: These examples only show entries for Assertion Attribute, Type, and Value.

Transformation Example 1

The following example assumes the "title" attribute is already in the assertion. The table indicates the user attributes in the user store.

User Directory Attributes	Attribute Value
role	admin
admintitle	SeniorAdmin
supertitle	SuperUser

Transform the value of the existing title attribute using the following configuration:

Assertion Attribute

title

Type

Expression

Value

#{attr[“role”] == ‘admin’ ? attr[“admintitle”] : attr[“supertitle”]}

Result: The expression is conditional based on the "role" user attribute being set to "admin." Assuming this condition is met, the assertion attribute "title" is set to the value SeniorAdmin, the "admintitle" attribute. If the role is something other than "admin," the "title" attribute becomes SuperUser, the value of the "supertitle" attribute.

Transformation Example 2

The following example assumes that the ContactNo attribute is already in the assertion.

User Directory Attributes	Attribute Value
homephone	555-3344
mobile	555-8888

Transform the value of the existing title attribute using the following configuration:

Assertion Attribute

ContactNo

Type

Expression

Value

#{attr["homephone"] == '555-3344' ? attr["mobile"] : attr["homephone"]}

Result: The expression is conditional on the logged-in user having the "homephone" user attribute set to 555-3344. Assuming this condition is met, the assertion attribute is set to 555-8888, the value of the "mobile" attribute. If the condition is not met, the "homephone" value does not change.

Note: To configure an expression that uses session attributes, replace attr["*attribute_name*"] with session_attr[“*attribute_name*”]. For example:

#{session_attr[“att1”] == ‘admin’ ? session_attr[“attr2”] : attr[“attr3”]}

Add a Claim to an Assertion

You can add an assertion attribute that is not already present.

Addition Example 1

The following example assumes the "title" assertion attribute is *not* in the assertion.

User Directory Attribute	Attribute Value
role	admin
admintitle	director
supertitle	executive

The following configuration adds the title attribute to the assertion.

Assertion Attribute

title

Type

Expression

Value

#{attr[“role”] == ‘admin’ ? attr[“admintitle”] : attr[“supertitle”]}

Result: The expression is conditional on the logged-in user having the "role" attribute set to admin. Assuming this condition is met, the assertion attribute "title" is added to the assertion and set to the value "director," the "admintitle" attribute value. If the role is something other than "admin," the assertion attribute "title" is added but the value becomes "executive," the value of the "supertitle" attribute.

Addition Example 2

The following example assumes the "smtitle" assertion attribute is *not* in the assertion.

User Directory Attribute	Attribute Value
title	manager

Assertion Attribute

smtitle

Type

Expression

Value

#{attr["title"] == 'manager' ? 'federation administrator' : attr["title"]}

Result: If the logged-in user has the title of "manager," add "smtitle" to the assertion and set its value to "federation administrator." You can enter any static value after the question mark instead of using the syntax attr["attribute_name"]. In this example, the static value is federation administrator,

Note: To configure an expression that uses session attributes, replace attr["*attribute_name*"] with session_attr[“*attribute_name*”]. For example:

#{session_attr[“att1”] == ‘admin’ ? session_attr[“attr2”] : attr[“attr3”]}

Delete a Claim from an Assertion

You can delete an assertion attribute.

Deletion Example 1

Delete the admintitle and supertitle assertion attributes by configuring two entries.

User Directory Attribute	Attribute Value
role	admin or superuser

User Directory Attribute	Attribute Value
title	administrator
su	superuser

Assertion Attribute

admintitle

Type

Expression

Value

#{attr[“role”] == ‘superuser’ ? ‘DELETE’ : attr[“title”]}

Result: The expression string is conditional based on the "role" user attribute. If the logged-in user has the role of superuser, delete the assertion attribute "admintitle." If the role is not superuser, set the title assertion attribute to the value of administrator, the value of the title user directory attribute.

Assertion Attribute

supertitle

Type

Expression

Value

#{attr[“role”] == ‘admin’ ? ‘DELETE’ : attr[“su”]}

Result: The expression string is conditional on the "role" user attribute. If the logged-in user role is "admin," delete the assertion attribute "supertitle." If the role is not "admin," set the supertitle assertion attribute to the value of superuser, the value of the su user directory attribute.

Deletion Example 2

The following example combines an addition and a deletion using one expression.

User Directory Attribute	Attribute Value
title	manager

Assertion attribute

ManagerName

Type

Expression

Value

#{attr["title"] != 'Manager' ? attr["manager"] : 'DELETE'}

Result: If the logged-in user does *not* have the user attribute title "manager," add the ManagerName attribute to the assertion. However, if the logged-in user title is manager, delete the ManagerName attribute, assuming it is part of the assertion.

Note: To configure an expression that uses session attributes, replace attr["*attribute_name*"] with session_attr[“*attribute_name*”]. For example:

#{session_attr[“att1”] == ‘admin’ ? session_attr[“attr2”] : attr[“attr3”]}

Customize Assertion Content

You can modify the assertion content using an assertion generator plug-in. The plug-in enables you to customize the content of an assertion based on the business agreements between you and your partners and vendors. One plug-in is allowed for each partner.

There are several steps to configuring an assertion generator plug-in.

1. Install the CA SiteMinder® Federation Standalone SDK, if you have not done so already.
2. Implement the AssertionGeneratorPlugin.java interface, which is part of the CA SiteMinder® Federation Standalone SDK.
3. Deploy your assertion generator plug-in implementation class.
4. Configure the plug-in the assertion generator plug-in parameters in the Administrative UI.

Implement the AssertionGeneratorPlugin Interface

The first step in creating a custom assertion generator plug-in is to implement the AssertionGeneratorPlugin interface. The following requirements apply to the implementation class:

- The implementation must provide a public default constructor method that contains no parameters.
- The implementation must be stateless, so that many threads can use a single plug-in class.
- The implementation must include a call to the customizeAssertion methods. You can overwrite the existing implementations of these methods as your requirements dictate. See the sample programs.
- The syntax requirements and use of the parameter string that is passed into the customizeAssertion method is the responsibility of the custom object.

Note: The folder *federation_sdk_home*\sample\com\ca\federation\sdk\plugin\sample includes two sample implementation classes.

Deploy an Assertion Generator Plug-in

After you have coded your implementation class for the AssertionGeneratoPlugin interface, compile it and verify that CA SiteMinder® Federation Standalone can find your executable file.

Follow these steps:

1. Compile the assertion plug-in code in one of the following ways:
   - If you are using a sample plug-in, use the build script for your platform to compile the plug-in. The build scripts are installed in the directory *federation_sdk_home*\sample. The build scripts are:

     Windows: build_plugin.bat

     UNIX: build_plugin.sh

     A compiled sample plug-in, fedpluginsample.jar, is in the directory *federation_sdk_home*\jar.
   - If you write your own plug-in, include the smapi.jar when you compile your plug-in.

2. In the JVMOptions.txt file, modify the -Djava.class.path value so it includes the classpath for the plug-in. Locate the JVMOptions.txt file in the directory *federation_install_dir*\siteminder\config.

   You can place the plug-in jar in any directory and have the JVMOptions.txt file point to it. To use the sample plug-in, modify the classpath to point to fedpluginsample.jar; however, do not modify the classpath for smapi.jar.

   Note: To use Apache Xerces or Xalan in your plug-in, use the Xerces or Xalan binary files installed with the product. The binaries are not installed with the federation SDK. Using these files is necessary for compatibility reasons.

3. Restart the CA SiteMinder® Federation Standalone services.

   Restarting the services helps ensure that CA SiteMinder® Federation Standalone uses the latest version of the assertion generator plug-in.

Enable the Assertion Generator Plug-in

After writing an assertion generator plug-in and compiling it, you enable the plug-in by configuring settings in the Administrative UI. The UI parameters let CA SiteMinder® Federation Standalone know where to find the plug-in.

Do not configure the plug-in settings until you deploy the plug-in.

Follow these steps:

1. Log on to the Administrative UI.
2. Navigate to the Assertion Configuration step of the Partnership wizard for the partnership you want to modify.

3. Enter values for the Assertion Generator Plug-in settings that follow:

Plug-in Class

Specifies the Java class name of the plug-in. Enter a name. This plug-in is invoked at run time.

Example: com.mycompany.assertiongenerator.AssertionSample

The plug-in class can parse and modify the assertion, and then return the result to CA SiteMinder® Federation Standalone for final processing. Specify an Assertion Generator plug-in for each relying party. A compiled sample plug-in is included in the SDK. You can view complied sample assertion plug-ins in the directory *federation_sdk_home*/jar.

Note: You can also view the source code for the CA SiteMinder® Federation Standalone sample plug-ins in the directory *federation_sdk_home*\sample\com\ca\federation\sdk\plugin\sample.

Plug-in Parameter

(Optional). Specifies the string that CA SiteMinder® Federation Standalone passes to the plug-in as a parameter at run time. The string can contain any value; there is no specific syntax to follow.

The plug-in interprets the parameters that it receives. For example, the parameter could be the name of an attribute or the string can contain an integer that instructs the plug-in to do something.

Reference information (method signatures, parameters, return values, data types), and the constructor for UserContext class and the APIContext class, are in the *Javadoc Reference*. Refer to the AssertionGeneratorPlugin interface in the Javadoc.

Chapter 11: Assertion Processing Customization (Relying Party)

This section contains the following topics:

Customize Assertion Processing (Relying Party) (see page 177)

Customize Assertion Processing (Relying Party)

The message consumer plug-in is a Java program that implements the Message Consumer Extension API. 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.

During authentication, the system first tries to process the assertion by mapping a user to its local user store. If CA SiteMinder® Federation Standalone cannot find the user, it calls the postDisambiguateUser method of the message consumer plug-in.

If the plug-in successfully finds the user, the process continues 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, the system calls the postAuthenticateUser method of the message consumer plug-in, if the plug-in is configured. If the method succeeds, CA SiteMinder® Federation Standalone 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.

Reference information (method signatures, parameters, return values, data types), and the constructor for UserContext class, are in the *Java SDK Programming Reference*. Refer to the MessageConsumerPlugin interface.

To configure the plugin:

1. Install the CA SiteMinder® Federation Standalone SDK.
2. Implement the MessageconsumerPlugin.java interface, which is part of the 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 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 outcome of assertion processing, regardless of whether the Policy Server processing is a success or failure.

The product provides the following samples of the Message Consumer plug-in class:

- MessageConsumerPluginSample.java
- MessageConsumerSAML20.java

The default location for the samples is:

Windows

C:\Program Files\Federation Standalone\sdk\java\sample

The package name is com\ca\federation\sdk\plugin\sample.

UNIX

/FederationStandalone/sdk/java/sample

The package name is com/ca/federation/sdk/plugin/sample.

Enable the Message Consumer Plug-in in the UI

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® Federation Standalone where to find the plug-in.

Do not configure the plug-in settings until you deploy the plug-in.

Follow these steps:

1. Log on to the Administrative UI.

   Select the Consumer-to-Producer or SP-to-IdP partnership that you want to modify.
2. Navigate to the User Identification step in the partnership wizard.
3. In the Message Consumer Plug-in section, complete the following fields:

   Plug-in Class

   Specify the Java class name for the plug-in, For example, a sample class included with the SDK is:

   com.ca.messageconsumerplugin.MessageConsumerPluginSample

   Plug-in Parameters

   Specify a string of parameters that are passed to the plug-in specified in the Full Java Class Name field.

4. Restart the federation services according to your operating environment.

   - **Windows**

     Use the stop and start shortcuts as follows. If you logged in as a network user and not a local administrator, right-click the shortcut and select Run as administrator.

     a. Start, All Programs, CA, Federation Standalone, Stop services

     b. Start, All Programs, CA, Federation Standalone, Start services

   - **UNIX**

     a. Open a command window.

     b. Run the following scripts:

     *federation_install_dir/*fedmanager.sh stop

     federation_install_dir/fedmanager.sh start

     Note: Do not stop and start the services as the root user.

Deploy a Message Consumer Plug-in

After you have coded your implementation class for the MessageConsumerPlugin interface, compile it and verify that CA SiteMinder® Federation Standalone can find your executable file.

Follow these steps:

1. Compile the MessageConsumerPlugin Java file. The file requires the following dependent libraries, which are installed with the product:

   federation_install_dir\siteminder\bin\jars\SmJavaApi.jar

   federation_install_dir is the directory where you installed CA SiteMinder® Federation Standalone

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 *federation_install_dir*\siteminder\config.

   Note: Do not modify the classpath for the existing xerces.jar, xalan.jar, or SmJavaApi.jar.

3. Restart the system 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.

Chapter 12: Single Sign-on Configuration

This section contains the following topics:

Single Sign-on Configuration (Asserting Party) (see page 181)
Single Sign-on Configuration (Relying Party) (see page 185)
Assertion Validity for Single Sign-on (see page 186)
Session Validity at a Service Provider (see page 188)
Status Redirects for HTTP Errors (SAML 2.0 IdP) (see page 189)
Single Sign-on Initiation (SAML 2.0) (see page 189)
Back Channel Authentication for Artifact SSO (see page 190)
How to Enable SAML 2.0 Attribute Query Support (see page 192)
How to Retrieve User Attribute Values from a Third-Party Source (see page 195)
How to Get User Consent to Send an Assertion (see page 199)
Enhanced Client or Proxy Profile Overview (SAML 2.0) (see page 203)
IDP Discovery Profile (SAML 2.0) (see page 206)
How to Configure SAML 2.0 HTTP POST Binding (see page 209)

Single Sign-on Configuration (Asserting Party)

When you configure single sign-on at the asserting party, you specify how the asserting party delivers an assertion to a relying party.

Only one single sign-on session is persisted in a browser. The session information is stored in the FEDSESSION cookie. If you access another partnership in the same browser, the FEDSESSION cookie is not valid, unless the underlying user directory is the same as the previously accessed partnership during the same browser session.

The FEDSESSION cookie uses the following timeout settings:

- Idle Timeout: 3600 seconds (1 hour)
- Max Timeout: 7200 seconds (2 hours)

You cannot change these timeout settings in UI.

Follow these steps:

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

   SAML 1.1

   Single Sign-On

   SAML 2.0

   SSO and SLO

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

2. Select an option for the Authentication Mode in the Authentication group box.

   Authentication Mode

   Select Local or Delegated

   - Click Local if the federation system is handling user authentication.
   - Click Delegated if a third-party web access management (WAM) system is handling user authentication.

3. Select the Authentication Type for the authentication mode you chose. The options change depending on whether you are using local or delegated authentication.

   Local Authentication Type (Local Mode only)

   Select Basic or Form based

   If you are using CA SiteMinder® Federation Standalone that is localized for Japanese or French users, select Forms based authentication scheme. Basic authentication is not supported for localized users.

   For forms authentication, sample log-in forms are available for Japanese and French. The forms are in the directory *federation_install_dir*/secure-proxy/proxy-engine/examples in the folders formsja (Japanese) and formsfr (French).

   To use the localized forms

   a. Navigate to *federation_install_dir/*secure-proxy/proxy-engine/examples.

   b. Make a backup copy of the forms folder.

   c. Rename the folder for your language (formsja for Japanese or formsfr for French) to **forms**.

   Delegated Authentication Type

   Select Legacy Cookie, Query String, Open-format Cookie

   Note: The open format cookie is the only FIPS-compatible option for delegated authentication.

4. For Delegated Authentication only, configure the required parameters for the type of delegated authentication you chose.

   Legacy Cookie

   If user identity information is being passed from the third-party WAM in a cookie, configure the Delegated Authentication URL. This URL redirects the request to the WAM system if the user comes to CA SiteMinder® Federation Standalone first. The URL does not apply when the user visits the WAM first.

   Query String

   If user identity information is being passed from the third-party WAM in a query string, configure the following settings:

   - Delegated Authentication URL

     This URL redirects the request to the WAM system when the user comes to CA SiteMinder® Federation Standalone first. The URL does not apply when user goes to the WAM first.

   - Hash Secret
   - Confirm Hash Secret

   Open-format Cookie

   If user identity information is being passed from the third-party WAM in a FIPS-encrypted cookie, configure the Delegated Authentication URL. The open format cookie is the only FIPS-compatible option for delegated authentication. This URL redirects the request to the WAM system if the user comes to CA SiteMinder® Federation Standalone first. The URL does not apply when user goes to the WAM first.

   Note: If you select Legacy Cookie or Open-format Cookie as the Delegated Authentication Type, configure the required global cookie settings. Locate the deployment settings by navigating to Infrastructure, Deployment Settings.

5. Complete the Authentication Class field by entering a URI for the user authentication method you want to use. This URI is placed in the AuthnContextClassRef element in the assertion to describe how a user is authenticated.

   Guidelines:

   - If the user is going to authenticate locally, accept the default URI for Password.
   - If the user is going to authenticate at a remote third party, edit this field to reflect the authentication method.

6. Complete the required fields in the SSO group box to configure how single sign-on operates:

   Be aware of the following guidelines:

   - If you select Artifact binding, select an artifact encoding (URL or FORM). The encoding defines how the artifact comes back to the relying party. If you select the URL option, the artifact is sent back as a query parameter in a URL. If you select FORM, the artifact is posted as form data.
   - You can select both bindings for SAML 2.0—the local entity determines the sequence in which the bindings are tried.

     Note: For artifact binding, the assertion is sent over a secure back channel. Therefore, configure the settings in the Back Channel group box.
   - When you select an SSO binding, configure at least one Assertion Consumer Service with a matching binding. If you select Enhanced Client and Proxy Profile, you need an Assertion Consumer Service with the PAOS binding.
   - The SSO Validity Duration and the Skew Time determine when the assertion is valid. Read the information about assertion validity (see page 186) to understand how these settings work together.

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

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

This procedure completes SSO configuration for the asserting party.

More information:

Assertion Validity for Single Sign-on (see page 186)
Delegated Authentication (see page 227)

Customize the Auto-POST form for HTTP-POST SSO

You can customize the auto-POST form sent to the relying party in a SAML response to improve the user experience.

To use a customized form, enter the name of the form in the Custom Post Form field of the SSO section of the SSO and SLO step of the wizard. The system uses the form you specify in the response. The product includes a form named defaultpostform.html.

Note: Enter only the name of the form, not the path to the form.

The physical page must reside in the directory *federation_install_dir*\customization, where *federation_install_dir* is the installed location of the product.

Authentication Options using Partnership Federation

Standalone partnership federation lets you select the authentication mode for federated single sign-on. You select the mode as part of the single sign-on configuration at the asserting party.

- **Local authentication mode**

  Local authentication happens at the local federation system. For local authentication, you can select Basic or Forms as the authentication schemes. These options are the only two methods available locally.

- **Delegated authentication mode**

  Delegated authentication forwards the authentication task to a third-part web access management (WAM) system. The method by which the third party authenticates a user depends on the authentication schemes the third party supports. After the third-party WAM authenticates the user, it returns the federated user identity to the entity originally asked to authenticate the user.

Single Sign-on Configuration (Relying Party)

To configure single sign-on at the relying party, you specify the SAML binding supported by the relying party and the related aspects of how the relying party handles single sign-on communication.

When CA SiteMinder® Federation Standalone is at the relying party, it uses the skew time set for the partnership to determine if the assertion it receives is valid. Read more about assertion validity (see page 186) to understand how CA SiteMinder® Federation Standalone uses the configured skew time.

Follow these steps:

1. Begin at the appropriate step in the Partnership Wizard.

   SAML 1.1

   Single Sign-On

   SAML 2.0

   SSO and SLO

2. Configure the settings in the SSO group box for the profiles you are using.

   For SAML 2.0, you can select both Artifact and POST—the local entity determines the sequence in which the bindings are tried.

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

3. If you select HTTP-Artifact, also configure the authentication method for the outgoing back channel.

This procedure completes the SSO configuration for the relying party.

Assertion Validity for Single Sign-on

For Single Sign-on, the values of the Skew Time and the SSO Validity Duration determine how CA SiteMinder® Federation Standalone calculates the total time that an assertion is valid. CA SiteMinder® Federation Standalone applies the skew time to the generation and consumption of assertions. In the assertion document, the beginning and end of the validity interval is represented by the NotBefore and NotOnOrAfter values.

At the asserting party, CA SiteMinder® Federation Standalone sets the assertion validity. CA SiteMinder® Federation Standalone determines the beginning of the validity interval by taking the system time when the assertion is generated. It sets the IssueInstant value in the assertion based on this time. CA SiteMinder® Federation Standalone then subtracts the skew time value from the IssueInstant value. The resulting time becomes the NotBefore value.

NotBefore=IssueInstant - Skew Time

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

Calculating Assertion Validity if CA SiteMinder® Federation Standalone is at Both Sides of the Partnership

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

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

The initial part of the equation (2 x Skew Time + SSO 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 CA SiteMinder® Federation Standalone, the SSO Validity Duration is only set at the asserting party.

Example

Asserting Party

The values at the asserting party are as follows:

IssueInstant=5:00PM

SSO Validity Duration=60 seconds

Skew Time = 60 seconds

NotBefore = 4:59PM

NotOnOrAfter=5:02PM

Relying Party

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

Skew Time = 180 seconds (3 minutes)

NotBefore = 4:56PM

NotOnOrAfter=5:05PM

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

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

Session Validity at a Service Provider

You can manage the duration of the authentication session at the Service Provider. The SessionNotOnOrAfter attribute is an optional attribute that the IdP can include in the <AuthnStatement> of an assertion. The configuration for session validity is done at the IdP.

Note: The SessionNotOnOrAfter parameter is different from the NotOnOrAfter parameter, which determines how long the assertion is valid.

A third-party SP can use the value of the SessionNotOnOrAfter to set its own timeout values, helping to ensure that sessions are not too short. If a user session becomes invalid, the user has to reauthenticate at the Identity Provider.

Important! If CA SiteMinder® is acting as an SP, it ignores the SessionNotOnOrAfter value. Instead, a CA SiteMinder® SP sets session timeouts from the realm timeout that corresponds to the SAML authentication scheme protecting the target resource.

Follow these steps:

1. Log in to the Administrative UI.
2. Select the IdP->SP partnership you want to modify.
3. Navigate to the SSO and SLO step.
4. In the SSO section, select the option for the SP Session Validity Duration. If you select the customize option, you can select several options.

   Note: Click Help for a description of fields, controls, and their respective requirements.
5. Select the Confirm step after you complete your changes and click Finish.

Status Redirects for HTTP Errors (SAML 2.0 IdP)

For the Identity Provider, you can configure how CA SiteMinder® redirects a user when an HTTP 500, 400, or 405 error occurs. For example, a 403 error can occur because the URL in a request points to the wrong target. If this error occurs, the user is sent to the specified URL for further processing.

Select the redirect options as follows:

1. Navigate to the Status Redirect URL section of SSO and SLO dialog.
2. In the Status Redirect URL section, select the check box for the error conditions that prompt a redirect.
3. Enter the destination URL where CA SiteMinder® redirects the user.
4. For each URL, select the redirect method, 302 No Data or HTTP Post.

Redirect handling is configured.

Single Sign-on Initiation (SAML 2.0)

For SAML 2.0 partnerships, you can determine whether the IdP or the SP or both can initiate single sign-on. You can configure which transactions are allowed at each side of the partnership.

Consider how restricting the initiation of a transaction can impact other single sign-on features, such as exchanging user authentication context information.

Follow these steps:

1. Log in to the Administrative UI.
2. Select the SAML 2.0 partnership you want to edit.

3. Navigate to the SSO and SLO step of the partnership wizard.
4. In the Transactions Allowed field, select an option from the pull-down menu.
5. Skip to the Confirm step of the wizard and save your changes.

Back Channel Authentication for Artifact SSO

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

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

Securing the back channel using SSL involves:

1. Enabling SSL.

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

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

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

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

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

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

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

   The options for back channel authentication are:

   - Basic
   - Client Cert
   - NoAuth

   The Administrative UI help describes these options in detail.

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

Configure the HTTP-Artifact Back Channel

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

Consider the following limitation:

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

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

Follow these steps:

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

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

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

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

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

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

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

How to Enable SAML 2.0 Attribute Query Support

A CA SiteMinder® IdP supports the SAML 2.0 Assertion Query/Request profile and can respond to attribute queries. The IdP also extends the profile functionality by accepting queries for attributes not in the assertion or in the metadata. When the IdP receives an attribute query, the IdP first checks its user directory to find the attributes. If the attributes are not found, the Policy Server checks the session store. The session store can hold attributes from external Identity Providers, attributes collected from advanced authentication schemes, and other sources.

Note: Only the CA SiteMinder® IdP supports the query profile. A CA SiteMinder® SP as an attribute requester is only supported for the proxied attribute query feature (see page 195).

The IdP has all the user attributes that an SP can request in its metadata. An SP can obtain these attributes in two ways:

- Extract the set of attributes that are sent in an assertion.

  The Identity Provider assertion configuration determines the set of attributes included. Defining a subset of all the attributes limits the number of attributes to the most essential, which reduces processing overhead.

- Import the IdP metadata.

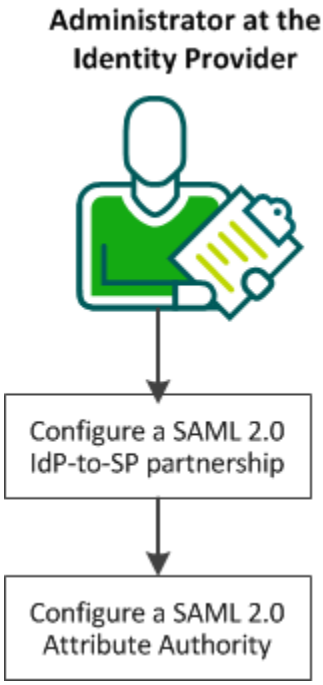
In addition to the attributes in the metadata, an SP can require attributes that are not in the assertion or in the metadata. To retrieve other attributes, the SP sends an attribute query to the IdP.

The query request profile employs two entities:

- SAML Attribute Authority
- SAML Attribute Requester

A CA SiteMinder® IdP can only act as an Attribute Authority. A CA SiteMinder® SP cannot be the Attribute Requester.

The following graphic shows the configuration steps for an Attribute Authority.



Complete the following steps:

- Configure or modify an IdP-to-SP partnership (see page 194).
- At the Identity Provider, configure a SAML 2.0 Attribute Authority (see page 194).

If CA SiteMinder® is at both sides of the partnership, you cannot use the Assertion Query/Response profile.

Configure the Partnership for Attribute Query Support

For the IdP to respond to attribute queries, an IdP-to-SP partnership must exist. You can create a partnership or modify an existing partnership.

The steps for creating a partnership include:

1. Create the SAML 2.0 IdP and SP entities (see page 107).
2. Configure a connection to a user directory for the partnership (see page 73).
3. Create a SAML 2.0 IdP-to-SP partnership (see page 143).
4. Configure a SAML 2.0 Attribute Authority (see page 194).

Configure the SAML 2.0 Attribute Authority

You can configure an IdP to serve as an Attribute Authority.

Follow these steps:

1. Log in to the Administrative UI.
2. Click Federation, Partnerships.
3. Select the IdP-to-SP partnership that you want to modify or create a new one.
4. Navigate to the SSO and SLO step of the partnership wizard.
5. Select Enable in the Attribute Service section of the dialog.
6. Enter a number of seconds for the Validity Duration.

   Note: Click Help for a description of fields, controls, and their respective requirements.
7. (Optional) Specify whether to require that the attribute query is signed, and the signing requirements for attribute assertions and responses.
8. Enter the search specifications for the appropriate user directory name space in the User Lookup section. The Attribute Authority uses this search specification to disambiguate the user.

   An example for an LDAP user directory is uid=%s. At least one search specification is required.

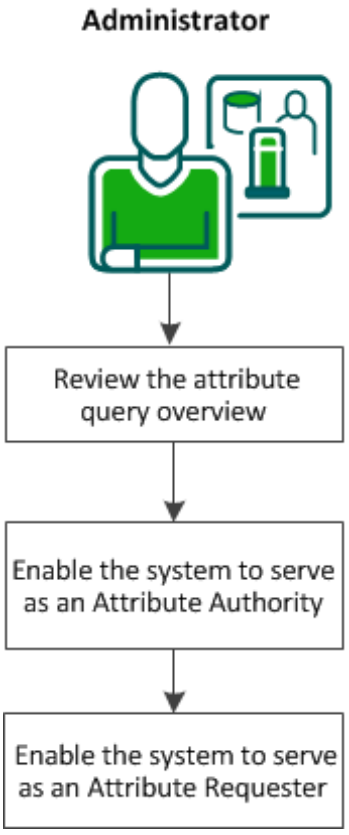
9. (Optional) Specify Partnership as the Protection Type in the Back Channel section. Select an authentication method. For more information about the back channel, click Help.
10. Save and activate the partnership.

The Identity Provider is now set up to serve as an Attribute Authority. This authority can now respond to attribute queries from a third-party SP.

How to Retrieve User Attribute Values from a Third-Party Source

In a SAML 2.0 federated environment, a Service Provider sometimes requires information about a user that is not provided in the assertion. The Service Provider can request the values of predetermined user attributes. If the Identity Provider does not have these values, it can request the values from a third party. In a CA SiteMinder® environment, this feature is referred to as a proxied attribute query.

The following diagram illustrates the process for enabling a proxied attribute query:



To enable a proxied attribute query, complete the following tasks:

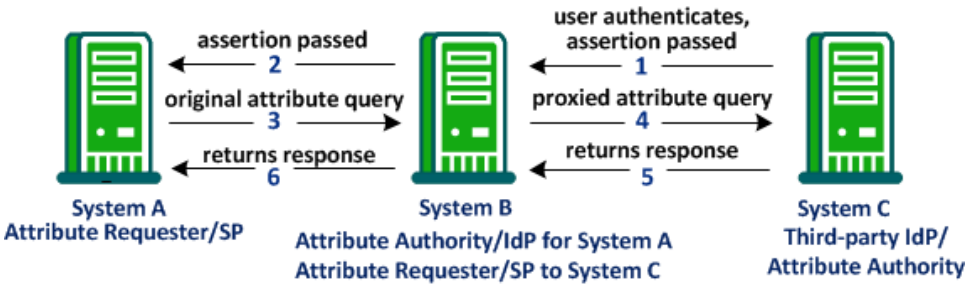
1. Review the proxied attribute query overview. (see page 196)
2. Enable the system to serve as an Attribute Authority (see page 197).
3. Enable the system to serve as an Attribute Requester (see page 198).

Proxied Attribute Query Overview

The proxied attribute query feature is based on the SAML 2.0 Assertion Query/Request profile and extends the search for user attributes. The Attribute Authority first searches the user directory and the session store for attributes. If the attribute is not found and the user initially authenticated at a third-party IdP, the request can be forwarded to the third-party IdP.

To implement a proxied attribute query, a single CA SiteMinder® system acts as a relay point between two remote systems. To relay the request from one remote system to another, the single system takes on two roles. The system first serves as the Attribute Authority for the original Attribute Requester. The system also serves as an Attribute Requester to the third-party IdP. As the Attribute Requester, the system proxies the attribute query to the original IdP.

The following figure shows how a single system processes the proxied query:



The following steps explain the flow of a proxied attribute query:

1. The user initially authenticates at the System C, the third-party IdP. System C generates an assertion and passes it to System B.
2. System B sends the assertion to System A, completing the initial single sign-on transaction between Systems A, B, and C. This single sign-on transaction is necessary to process a proxied attribute query.

3. After System A receives the assertion, the system determines that it needs other attributes that are not in the assertion. As the Attribute Requester, System A sends an attribute query to its Attribute Authority/IdP, System B.
4. System B determines that System A requires attributes that are not in its user directory or session store. To retrieve the attributes, System B generates a new query request. It sends the new query to System C, the third-party IdP, where the user originally authenticated. This new query is the proxied query.
5. System C returns a response with the attributes to System B. System B saves the attributes in its session store.
6. System B, in its role as the Attribute Authority, returns its own response with the attributes to System A.

Important! The configured attribute names and the name format (unspecified, uri, or basic) at System A must match the names of these attributes at System C. This information is communicated before any transactions occurs.

Enable the System to Serve as an Attribute Authority (IdP->SP)

To implement a proxied query transaction, configure two partnerships on the same CA SiteMinder® system:

- An IdP-to-SP partnership
- An SP-to-IdP partnership

For CA SiteMinder® to serve as an Attribute Authority, modify an existing IdP-to-SP partnership or create a partnership. In this partnership, CA SiteMinder® is the local IdP/Attribute Authority and the remote partner is the SP/Attribute Requester.

Note: This system also serves as the Attribute Requester in the SP-to-IdP partnership.

Follow these steps:

1. Log in to the Administrative UI.
2. Click Federation, Partnerships.
3. Select the IdP-to-SP partnership that you want to modify or create a new one.
4. Navigate to the SSO and SLO step of the partnership wizard.
5. Select Enable in the Attribute Service section of the dialog.
6. Enter a number of seconds for the Validity Duration.

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

7. (Optional) Specify whether to require that the attribute query is signed, and the signing requirements for attribute assertions and responses.
8. Select Enable Proxied Query.
9. Enter the search specifications for the appropriate user directory name space in the User Lookup section. The Attribute Authority uses this search specification to disambiguate the user.

   An example for an LDAP user directory is uid=%s. At least one search specification is required.
10. (Optional) Specify Partnership as the Protection Type in the Back Channel section. Select an authentication method. For more information about the back channel, click Help.
11. Save and activate the partnership.

The system can now serve as an Attribute Authority to the original Attribute Requester.

Enable the System to Serve as an Attribute Requester (SP->IdP)

To implement a proxied query transaction, configure two partnerships on the same CA SiteMinder® system:

- An IdP-to-SP partnership
- An SP-to-IdP partnership

Note: Partnership federation supports the SP as an Attribute Requester only for the proxied attribute query feature.

For CA SiteMinder® to serve as an Attribute Requester, modify an existing SP-to-IdP partnership or create a partnership. In this partnership, CA SiteMinder® is the local SP/Attribute Requester and the remote third party is the remote IdP/Attribute Authority.

Note: This system also serves as the Attribute Authority in the IdP-to-SP partnership.

Follow these steps:

1. Log in to the Administrative UI.
2. Click Federation, Partnerships.
3. Select the SP-to-IdP partnership that you want to modify or create a new one.
4. Navigate to the SSO and SLO step of the partnership wizard.
5. Select Enable and Enable Proxied Query in the Attribute Requester Service section.

6. Provide a URL for the remote IdP in the Attribute Services section.
7. Provide the format, type, and value for the Name ID.

   Note: Click Help for a description of fields, controls, and their respective requirements.
8. (Optional) Select an authentication type for the back channel. For information about the back channel, click Help.
9. Save and activate the partnership.

The Service Provider can now serve as an Attribute Requester.

How to Get User Consent to Send an Assertion

A federated partnership relies on trust between the two parties. Part of the trust relationship can be a contractual requirement to have user permission to pass on identity information to a relying partner. Additionally, users that control whether to exchange their identity information for a requested service helps enforce the trust relationship.

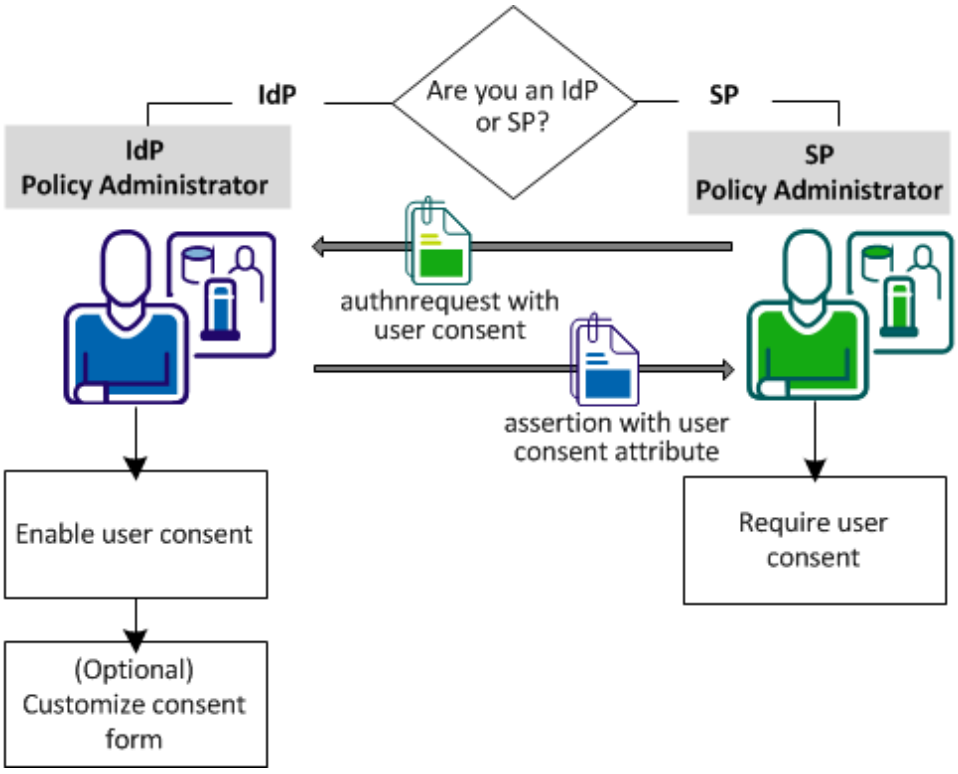
Your federation system acting as an Identity Provider supports the SAML 2.0 user consent feature. User consent at the Identity Provider site requires that the Identity Provider asks the user to grant permission before it sends an assertion to a partner. If you enable user consent at the Identity Provider, the Identity Provider prompts the user for consent. The Identity Provider passes the consent value in an assertion.

The consent validity period is 5 minutes. When the Identity Provider redirects the user to the consent page, the user has 5 minutes to grant consent. The user is then redirected back to the Identity Provider. The Identity Provider then generates the assertion and sends it to the Service Provider. These tasks must be complete in the 5-minute time period. If the time expires before the Identity Provider generates an assertion, it does not pass on the user identity.

Consent applies only to a single assertion. After the Identity Provider generates an assertion, it deletes all record of consent being granted. The same user can return to an Identity Provider before the 5-minute validity period expires, but the Identity Provider still prompts the user for consent.

Note: The validity period is not configurable.

This figure shows the configuration tasks at each partner.



The configuration tasks at the IdP are:

1. Enable user consent at the IdP (see page 201).
2. Customize a user consent form (optional) (see page 202).

The configuration task at the SP is:

1. Require user consent at the SP. (see page 203)

User Consent Example

The following use case illustrates user consent.

User1 logs in and authenticates at MyWorkPlace.com at 2:00PM. MyWorkPlace is acting as an Identity Provider. At 2:03PM, the user selects a link to the partner company that runs travel specials for employees. User1 is redirected to a form that asks for consent before sending User1 to ExampleTravel.com. User1 takes a phone call before completing the consent form. The time is now 2:10PM. MyWorkPlace does not generate an assertion because the validity period has expired.

If User1 grants consent promptly and is redirected back to the Identity Provider by 2:05PM, the Identity Provider generates an assertion. Only 2 minutes pass between consent and assertion generation, so the validity period is still active.

Enable User Consent at the IdP

Configuring user consent requires that you:

- Enable user consent in the Administrative UI.
- Provide the name of a user consent form.

  The Identity Provider sends the custom form to the user to get consent.

Configure user consent at the Identity Provider using the Administrative UI. When you configure this feature through the UI, only the following URI is used in the assertion response:

```
urn:oasis:names:tc:SAML:2.0:consent:obtained
```

You can also enable this feature using the CA SiteMinder® Federation Standalone Java or .NET SDKs. The SDK passes whatever user consent value it receives from the third party that is performing delegated authentication.

User consent is also configurable at the Service Provider. A Service Provider can require the Identity Provider to pass the user consent value in the assertion response.

1. Log in to the Administrative UI.
2. Navigate to Federation, Partnership Federation, Partnerships.
3. Select the IdP->SP partnership you want to modify.
4. Navigate to the SSO and SLO step in the partnership wizard.

5. In the SSO section:
   a. Select the Enable User Consent check box.
   b. Specify the name of the custom form in the User Consent Post Form field.

   Note: The User Consent Service URL is specified by default. You cannot change this value.
6. Navigate to the Confirm step when your configuration is complete and click Finish.

Customize a User Consent Form (Optional)

The product ships with a *consent to federate* form named ca_defaultconsentform.html. The Identity Provider sends the custom form to the user to get permission to send an assertion for that user. The default consent form is in the following locations:

Windows: %FEDROOT%\customization

UNIX: $FEDROOT/customization

FEDROOT is the system environment variable.

You can write a custom form instead of using the default consent form.

Follow these steps:

1. Create the custom HTML form. Modify the form and replace values for the following settings:

   $$userconsent_spid$$

   Represents the SP ID configured in the partnership

   $$userconsent_idpid$$

   Represents the IDP ID configured in the partnership.
2. Place the form in the customization directory.
3. Specify the location of the form User Consent Post Form in the Administrative UI.

Require User Consent at the SP

The SP can require that the user consent attribute be in the assertion response returned by the IdP. To include this attribute in the authentication request, enable the setting in the Administrative UI.

Follow these steps:

1. Log in to the Administrative UI.
2. Modify the appropriate SP->IdP partnership.
3. Navigate to the SSO and SLO step in the partnership wizard.
4. Select the Require User Consent setting in the SSO section of the dialog.

   Note: Click Help for a description of fields, controls, and their respective requirements.
5. If there are no other changes, select the Confirm step and click Finish to save the changes.

The user consent attributes is placed in the authentication request sent to the IdP.

Enhanced Client or Proxy Profile Overview (SAML 2.0)

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.

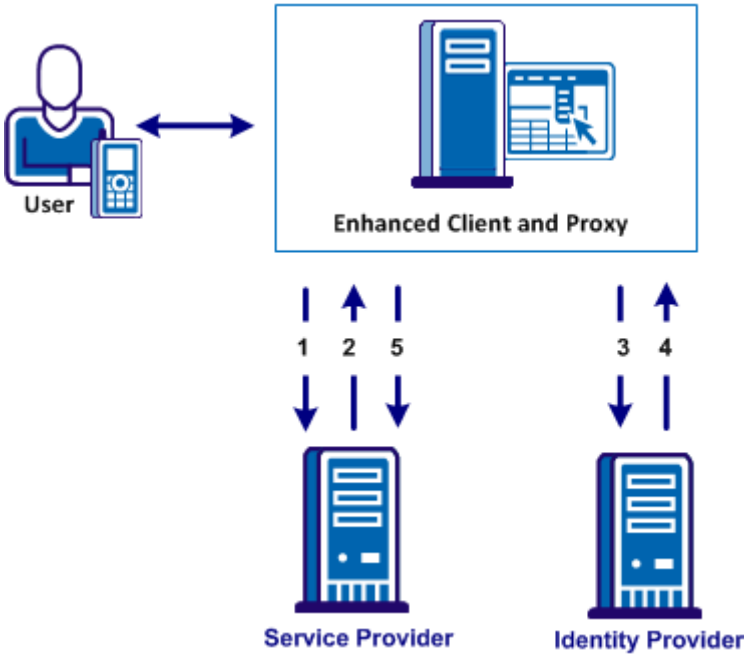
The ECP profile enables single sign-on when the Identity Provider and Service Provider cannot communicate directly. The ECP acts as the intermediary between the Service Provider and the Identity Provider.

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. CA SiteMinder® Federation Standalone only processes the 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 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 directory 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® Federation Standalone Identity Provider.

Follow these steps:

1. Log in to the Administrative UI.
2. Select the local Identity Provider partnership that you want to modify.
3. Navigate to the SSO and SLO step in the partnership wizard.
4. In the SSO section, select the Enable Enhanced Client or Proxy Profile check box.
5. Navigate to the Confirm step and click Finish to save changes.

The Identity Provider can now process ECP calls.

Note: A single 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.

Configure ECP at the Service Provider

To configure ECP, you must enable the feature at the Identity Provider and the Service Provider. The following procedure is for a Service Provider.

Follow these steps:

1. Direct the 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.
3. Modify the relevant local Service Provider partnership.
4. Navigate to the SSO and SLO step in the partnership wizard.
5. In the SSO section, select the Enable Enhanced Client or Proxy Profile check box.
6. Navigate to the Confirm step and click Finish to save the change.

The Service Provider can now process ECP calls.

Note: A single 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.

IDP Discovery Profile (SAML 2.0)

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.

IDP Discovery Configuration at the Identity Provider

You configure the IDP Discovery profile in the IDP Discovery section in the SSO and SLO dialog.

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

To enable the Identity Provider Discovery Profile

1. Select the Enable IDP Discovery checkbox.
2. Set the value for the Service URL field to the Identity Provider Discovery Profile servlet. For CA SiteMinder® Federation Standalone, this URL is:

   http://*host:port*/affwebservices/public/saml2ipd

   host

   Represents the common domain that you specify in the Common Domain field.

   port

   Specifies the Apache HTTP or HTTPS port you specified when installing CA SiteMinder® Federation Standalone.

   The URL can also begin with https.
3. Specify the cookie domain in the Common Domain field.
4. (Optional) Select the Enable Persistent Cookie check box to preserve the common cookie in the browser.

IdP Discovery is enabled at the IdP.

IDP Discovery Configuration at the Service Provider

For IDP Discovery profile, the Service Provider (SP) has to determine the Identity Provider (IdP) to which it sends authentication requests. The user that the SP wants to authenticate must have previously visited the Identity Provider and authenticated.

The SP has to redirect the user to its own IdP Discovery Service to retrieve the common domain cookie. The cookie contains the list of Identity Providers that the user has already visited. From this list, the cookie chooses the correct IdP and then sends an AuthnRequest to that IdP.

The IDP Discovery process is 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 IDP Discovery Service URL, indicating that it wants to get the Common Domain Cookie.

3. The IDP Discovery Service gets the Common Domain Cookie, reads the cookie in its domain and redirects the user back to the site selection page. The discovery service provides Common Domain Cookie as a query parameter.

4. The SP populates the site selection page with IdP URLs at which the user has previously authenticated.

5. The user selects an IdP to perform the user authentication.

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.

   CA SiteMinder® Federation Standalone 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:

   federation_install_dir/secure-proxy/Tomcat/webapps/affwebservices/public

   The first link redirects the browser from one domain to the IdPDiscovery service in the common domain and retrieves the common domain cookie, named **_saml_idp**. When the IdP Discovery Service at the SP receives the request, it gets the common domain cookie and adds it as a query parameter. The IDP Discovery 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. This list is static; there are no HTML links associated with the list that initiate communication with the associated IdP.

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</a>
   ```

   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, you can see the site selection page with a list of IdPs from which to select.

Securing the IdP Discovery Target Against Attacks

When the CA SiteMinder® Federation Standalone 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 and 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 used by the Web Agent, but this setting is defined specifically for federation.

When the IPD Service examines the IPDTarget query parameter, it obtains the domain of the URL specified by the query parameter. The IPD Service compares this domain to the list of domains specified for 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, no validation is done and the user is redirected to the target URL.

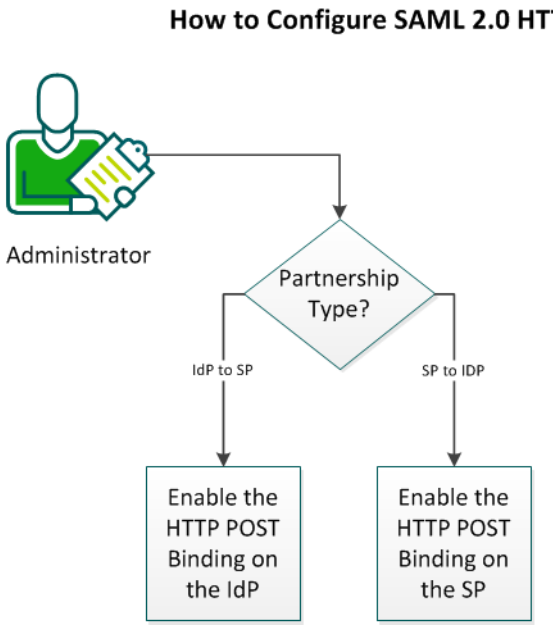
How to Configure SAML 2.0 HTTP POST Binding

For authentication and single log-out requests, you can enable SAML 2.0 HTTP POST binding as a method for exchanging requests and responses.

This procedure assumes that you are familiar with federated environments and have created and activated one or more of the following partnerships:

- IdP to SP
- SP to IdP

The following graphic describes how to enable SAML 2.0 HTTP POST binding:



Follow these steps:

1. Perform the appropriate task for your type of partnership:
   - Enable the HTTP POST binding at the IdP (see page 210).
   - Enable the HTTP POST binding at the SP (see page 211).

Enable the HTTP POST Binding at the IdP

You can enable the HTTP POST binding on the IdP side (of an IdP to an SP partnership).

Follow these steps:

1. Open the Administrative UI.
2. If the partnership that you want to modify is active, deactivate it.
3. Click Modify to open the partnership wizard.
4. Navigate to the SSO and SLO tab in the partnership wizard.
5. Click the HTTP POST check boxes next to the following items:

   Authentication Request Binding

   Specifies the types of bindings the IdP allows when it receives an authentication request from the SP.

   Options: HTTP-Redirect, HTTP-POST

   SSO Profile

   Determines which single sign-on profile the federation system uses for processing requests. You can select all bindings; the local entity determines the sequence in which the bindings are tried.

   Options: HTTP-Artifact, HTTP-POST, Enhanced Client and Proxy

   Note: Select the ECP profile if the entities in the partnership are communicating indirectly through an enhanced client. An enhanced client can be a browser or other user agent, or an enhanced proxy, such as a wireless proxy for a wireless device.

   Note: You can select multiple bindings for each item. For example, you can select HTTP-Redirect and HTTP-POST together. You also can select different bindings for each of these items. For example, you can select the HTTP-POST binding for one and the HTTP-Redirect binding for another.

6. Create a remote assertion consumer service URL with a binding that matches the one in your SSO profile. For example, if you picked HTTP-Redirect and HTTP-POST bindings, create two remote assertion consumer service URLs. One URL for each binding.
7. (Optional) In the SLO section, select the HTTP-POST check box.

   Note: You can select multiple bindings for this item. For example, you can select HTTP-POST and HTTP-Redirect together.
8. Create a SLO service URL with a binding that matches the SLO binding. For example, if you picked HTTP-Redirect and HTTP-POST bindings, create two SLO service URLs. One service URL for each SLO binding.
9. Complete any other partnership information as needed.
10. At the confirm step, click Finish.

    SSO HTTP-POST binding is enabled.

Enable the HTTP POST Binding at the SP

You can enable the HTTP POST binding at the SP side (of an SP to an IdP partnership).

Follow these steps:

1. Open the Administrative UI.
2. If the partnership that you want to modify is active, deactivate it.
3. Click Modify to open the partnership wizard.
4. Navigate to the SSO and SLO tab in the partnership wizard.

5. Click the HTTP POST check boxes next to the following items:

   Authentication Request Binding

   Specifies the types of bindings the SP uses when it sends an authentication request to the IdP.

   Limits: HTTP-Redirect, HTTP-POST

   SSO Profile

   Determines which single sign-on profile the federation system uses for processing requests. You can select all bindings; the local entity determines the sequence in which the bindings are tried.

   Options: HTTP-Artifact, HTTP-POST, Enhanced Client and Proxy

   Note: Select the ECP profile if the entities in the partnership are communicating indirectly through an enhanced client. An enhanced client can be a browser or other user agent, or an enhanced proxy, such as a wireless proxy for a wireless device.

   Note: You can select multiple bindings for each item. For example, you can select HTTP-Redirect and HTTP-POST together. You also can select different bindings for each of these items. For example, you can select the HTTP-POST binding for one and the HTTP-Redirect binding for another.

6. Create a remote SSO service URL with a binding that matches the one in your SSO profile. For example, if you picked HTTP-Redirect and HTTP-POST bindings, create two remote assertion consumer service URLs. One URL for each binding.
7. (Optional) In the SLO section, select the HTTP-POST check box.

   Note: You can select multiple bindings for this item. For example, you can select HTTP-POST and HTTP-Redirect together.

8. Create an SLO Service URL with a binding that matches the SLO binding. For example, if you picked HTTP-Redirect and HTTP-POST bindings, create two SLO Service URLs. One URL for each SLO binding.
9. Complete any other partnership information as needed.
10. At the confirm step, click Finish.

    SSO HTTP-POST binding is enabled.

Chapter 13: Configure Social Sign-on

You can configure CA SiteMinder® Federation Standalone (the federation system) to let users sign-on to a federated resource with their social networking credentials instead of the federation system credentials.

The social sign-on feature consists of the following features:

- Authentication of users using an OAuth authorization server such as Facebook so that users can sign-on to a federated resource using their OAuth authorization server credentials.
- Configuration of a credential selector page that provides users with various identity providers such as SAML 2.0 or Facebook as authentication choices. Users can choose an identity provider for authorization to sign-on to a federated resource.

The features are independent of each other and you can configure the federation system to implement either or both the features.

Authenticate Users Using an OAuth Authorization Server

To authenticate users using an OAuth authorization server, configure single sign-on between the federation system and the OAuth authorization server.

The federation system provides default support for the following OAuth authorization servers:

OAuth 1.0a

- Twitter

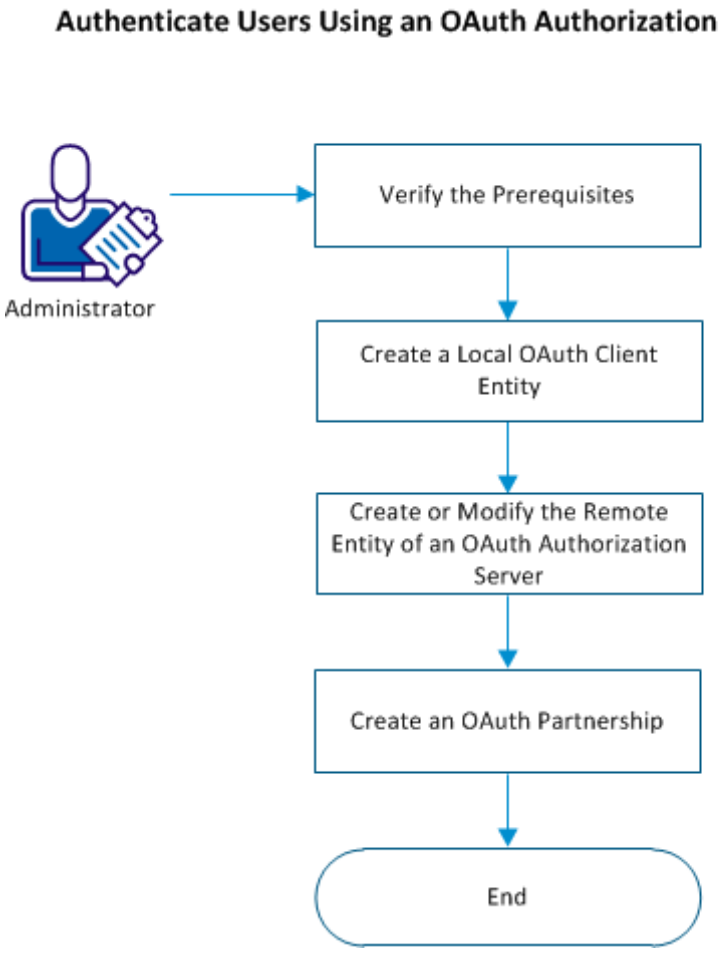
OAuth 2.0

- Facebook
- Google
- LinkedIn
- Windows Live

The following process describes how the federation system processes a user request to access a federated resource:

1. The federation system redirects the user request to the OAuth authorization server specified in the user request.
2. The OAuth authorization server authenticates the user and sends an authentication response with claims about the user to the federation system.
3. The federation system verifies the authentication response, completes the authentication process, and authorizes the user to access the federated resource.

The following flowchart describes how you can authenticate users using an OAuth authorization server:



Follow these steps:

1. Verify the prerequisites.
2. Create a local OAuth client entity.
3. (Optional) Create or modify the remote entity of the OAuth authorization server.
4. Create an OAuth partnership between the federation system and the OAuth authorization server.

Verify the Prerequisites

Perform the following steps before you configure a partnership to configure single sign-on between the federation system and an OAuth authorization server:

- Enable SSL on the federation system.
- To use an OAuth authorization server that the federation system support by default, perform the following steps before you invoke the partnership:
  - On a standalone deployment, verify that the default CA certificate of the OAuth authorization server was imported.
  - On an integrated deployment, import the default CA certificate of the OAuth authorization server using smkeytool.
- To use an OAuth authorization server that the federation system does not support by default, obtain and import the SSL CA certificate of the OAuth authorization server before you invoke the partnership.

Create a Local OAuth Client Entity

Create a local OAuth client entity for the partnership between the federation system and an OAuth authorization server.

Follow these steps:

1. Navigate to Federation, Entities, and click Create Entity.
2. Choose Local in Entity Location.
3. Select OAuth Client from New Entity Type.
4. Select the OAuth version, and click Next.
5. Enter the required values, and click Next.
6. Confirm the entered values and click Finish.

   The Redirect URL is constructed. Use this URL for initiating an OAuth transaction.

Create or Modify the Remote Entity of an Authorization Server

The system provides remote entities for each of the following OAuth authorization servers that are supported by default:

- OAuth 1.0a
  - Twitter
- OAuth 2.0
  - Facebook
  - Google
  - LinkedIn
  - Windows Live

The values of each remote entity are pre-configured with known values of the entity. You can modify the values to suit your federation environment or create a remote entity for any OAuth authorization server.

Follow these steps:

1. Perform *one* of the following tasks:
   - To create a new remote entity, perform the following steps:
     a. Navigate to Federation, Entities, Create Entity.
     b. Select Remote as Entity Location, and select OAuth Authz Server as the New Entity Type.
     c. Click Next.
     d. Enter the values and click Next.
   - To modify the pre-populated values of a remote entity, perform the following steps:
     a. Navigate to Federation, Entities, and search for the entity that you want to modify.
     b. Click the Actions option of the entity, and click Modify.
     c. Click Next to go to the Configure Entity tab.
     d. Modify the values and click Next.
2. Confirm the changes and click Finish.

Create a Partnership to Configure Single Sign-On

To let the federation system retrieve user information from the authorization server, create an OAuth partnership between the OAuth authorization server as the asserting party and the federation system as the relying party.

Follow these steps:

1. Navigate to Federation, Partnerships and click Create Partnership.
2. Select the OAuth Client - Authz Server partnership type.
3. Configure the partnership information.
4. Confirm the values and click Finish.

An OAuth partnership is configured to let users sign-on to a federated resource using the OAuth authorization server credentials.

When the federation system receives a user request in the following format, the request is processed per the partnership configuration:

https://<*baseURL_of_the_partnership*>/affwebservices/public/oauthtokenconsumer?AuthzServerID=<*authorization_server_id*>

Or

https://<*baseURL_of_the_partnership*>/affwebservices/public/oauthtokenconsumer/<disambiguation id>?AuthzServerID=<*authorization_server_id*>

The federation system is configured to implement the social sign-on feature.

Migrate an OAuth Authentication Scheme Set-up to OAuth Partnership

If you configured an OAuth authentication scheme in your environment to authenticate users using an OAuth provider, you can migrate your authentication scheme set-up to a federation partnership.

Follow these steps:

1. Perform one of the following steps:
   - If you want to use both the OAuth authentication scheme and an OAuth partnership simultaneously, register the application with an OAuth authorization server and add the new redirect URL of the following format to the existing OAuth authentication scheme redirect URL:

     `https://server:port/affwebservices/public/oauthtokenconsumer`
   - If you want to use an OAuth partnership instead of the OAuth authentication scheme, update the existing redirect URL at the OAuth authorization server to the appropriate partnership redirect URL of the following format:

     `https://server:port/affwebservices/public/oauthtokenconsumer`

     Note: The OAuth authentication scheme does not work after you update the authentication scheme redirect URL with the partnership redirect URL.
2. Create a partnership between the OAuth client and the OAuth authorization server.
3. Inform the application users that the following URL must be used to initiate the OAuth partnership:

   `https://server:port/affwebservices/public/oauthtokenconsumer?AuthzServerID=AuthorizationServerID`

Configure a Credential Selector Page

You can configure a partnership to let users select an identity provider such as Facebook or Twitter for authentication. The credential handling service that is installed on CA SiteMinder® SPS lets you configure a partnership to display a credential selector page with multiple identity providers as user authentication choices.

To configure a credential selector page, create the following partnerships:

1. Partnership to configure single sign-on between the federation system and an identity provider. The identity provider acts as the asserting party and the federation system acts as the relying party.
2. Partnership between the federation system and the enterprise where the federated resource exists. The federation system acts as the asserting party and the enterprise acts as the relying party.

The following process describes how the federation system processes a user request:

1. The enterprise (relying party) redirects the user request to the federation system (asserting party).
2. The federation system (asserting party) verifies if the partnership is configured to display a credential selector page. If yes, displays the credential selector page with multiple identity providers as user authentication choices.
3. The federation system performs *one* of the following tasks based on the user registration with the federation system:
   - If the user is registered in the federation system, the following steps are performed:
     a. The user selects an identity provider and signs-on with the identity provider.
     b. The identity provider generates an access token and redirects the user to the federation system (relying party).
     c. The federation system (relying party) verifies the access token and tries to disambiguate the user in the user store.
     d. The federation system (relying party) generates a session and redirects the user to the federation system (asserting party).
     e. The federation system (asserting party) generates an assertion and redirects the user to the enterprise (relying party).
     f. The enterprise (relying party) verifies the assertion and gives the user access to the federated resource.
   - If the user is not registered in the federation system, the following steps are performed:
     a. The user clicks the Register link.

b. The federation system displays the list of identity providers whose partnerships are configured with a provisioning server.

c. The user selects an identity provider and signs-on with the identity provider.

d. The identity provider generates an access token and redirects the user to the federation system (relying party).

e. The federation system (relying party) verifies the access token and tries to disambiguate the user in the user store.

f. The federation system (relying party) redirects the user to the provisioning server configured in the partnership.

g. The provisioning server creates the user and redirects the user to the federation system (relying party).

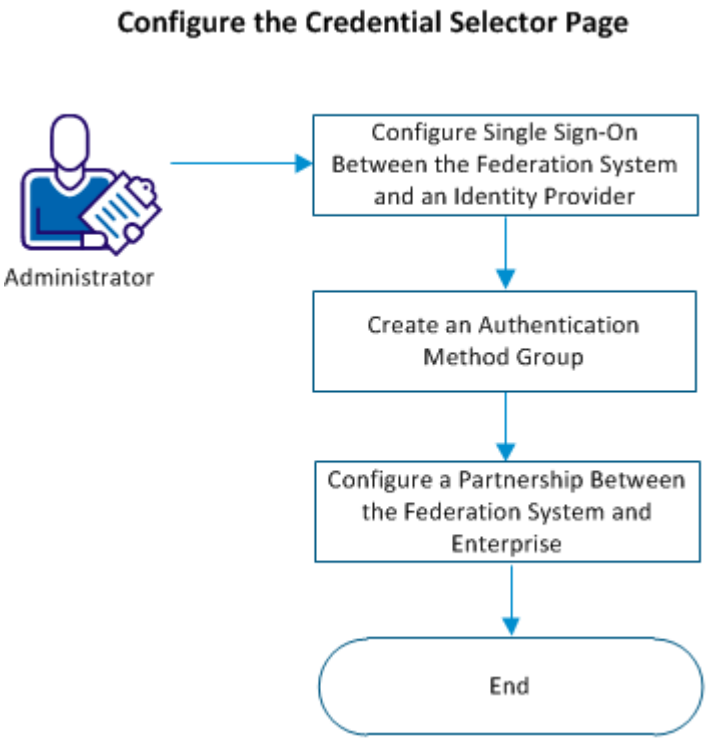
h. The federation system (relying party) generates a session and redirects the user to the federation system (asserting party).

i. The federation system (asserting party) generates an assertion and redirects the user to the enterprise (relying party).

j. The enterprise (relying party) verifies the assertion and gives the user access to the federated resource.

Thus, the user request is processed.

The following flowchart describes how to configure a credential selector page:



Follow these steps:

1. Configure single sign-on between the federation system and an identity provider.
2. Create an authentication method group.
3. Configure a partnership between the federation system and enterprise to display a credential selector page.

Configure Single Sign-On Between the Federation System and an Identity Provider

For each identity provider that you want to display on the credential selector page, create a partnership to configure single sign-on between the identity provider and the federation system. The identity provider acts as the asserting party and the federation system acts as the relying party.

The identity provider that acts as an authentication choice must be based on the following authentication protocols:

- SAML 1.1
- SAML 2.0
- WS-Federation
- OAuth

To let the federation system act as an identity provider, create a partnership with the system acting as both the asserting party and the relying party.

Follow these steps:

1. Navigate to Federation, Partnership.
2. Create a partnership for each identity provider that you want to display on the credential selector page.

Create an Authentication Method Group

An authentication method group defines the list of identity providers that must be displayed on the credential selector page. Each identity provider such as SAML or Facebook that you want to display on a credential selector page must be part of an authentication method group. When you create an authentication method group, you can select identity providers from a list of all the partnerships with identity providers acting as the asserting party.

Follow these steps:

1. Navigate to Federation, Authn Method Groups.
2. Click Create Authn Method Group.
3. Add the partnerships of the identity providers that you want to display as authentication choices, and enter the required values.
4. Save the changes.

Configure a Partnership Between the Federation System and Enterprise

Configure a partnership between the federation system and your enterprise to display the credential selector page when a user tries to access a federated resource. The federation system acts as the asserting party and the enterprise acts as the relying party. You can create a partnership or modify an existing partnership.

The partnership must be based on *one* of the following authentication protocols:

- SAML 1.1
- SAML 2.0
- WS-Federation

Follow these steps:

1. Navigate to Federation, Partnership.
2. Enter the values in each step.
3. Perform the following steps in the Single Sign-On, or SSO and SLO, or Single Sign-On and Sign-Out step:
   a. Select the Authentication Mode as Credential Selector.
   b. Define the Authentication Base URL.
   c. Select an Authentication Method Group.
4. Select the following field in the Target Application Configuration step:
   - SAML 1.1: Target
   - SAML 2.0 and WS-Federation: Relay State Overrides Target
5. Save the changes.

The partnership is configured to display the credential selector page when a user tries to access a federated resource.

The federation system is configured to implement the social sign-on feature.

Customize the Header and Footer on the Credential Selector Page

You can customize the header and footer that are displayed on the credential selector page to suit your enterprise requirements.

Follow these steps:

1. Navigate to the following location on your federation system:

   `<install_path>\CA\Federation Standalone\secure-proxy\Tomcat\webapps\chs\jsps`

2. Make a copy of the header.jsp file and name the new file as header-custom.jsp.
3. Make a copy of the footer.jsp file and name the new file as footer-custom.jsp.

   Note: If the header-custom.jsp and footer-custom.jsp files exist, the federation system is configured to use the files for displaying the header and footer.

4. Modify the files to customize the header and footer that must be displayed on the credential selector page.
5. Save the changes.
6. Restart CA SiteMinder® SPS.

   The customized header and footer are displayed on the credential selector page when the partnership is active.

Chapter 14: Delegated Authentication

This section contains the following topics:

Delegated Authentication Overview (see page 227)
How the Third Party WAM Passes the User Identity (see page 228)
Delegated Authentication Configuration (see page 234)

Delegated Authentication Overview

When you configure single sign-on for a federation partnership, one of your configuration decisions is determining how users are authenticated.

CA SiteMinder® Federation Standalone offers two authentication choices:

- Local authentication
- Delegated authentication

CA SiteMinder® Federation Standalone can perform local authentication; however, Basic and HTML forms are the only available authentication schemes.

Delegated authentication lets CA SiteMinder® Federation Standalone use a third-party web access management (WAM) system to perform the authentication of any user who requests a protected federated resource. The third-party WAM system performs the authentication and then forwards the federated user identity to CA SiteMinder® Federation Standalone. After CA SiteMinder® Federation Standalone receives the user identity information, it locates the user in its own user directory and starts the federation process with the relying party.

A delegated authentication request takes place at the asserting party and it can be initiated at the third-party WAM system or at CA SiteMinder® Federation Standalone. An authentication request can initiate at the relying party; however this is not considered delegated authentication.

Authentication can be initiated as follows:

Authentication Initiated by CA SiteMinder® Federation Standalone at the Asserting Party

CA SiteMinder® Federation Standalone can initiate an authentication request at an asserting party. If the request is made to CA SiteMinder® Federation Standalone, it is recognized as a delegated authentication request. CA SiteMinder® Federation Standalone then redirects the user to the third-party WAM system.

Authentication Initiated by Direct Login to the WAM System at the Asserting Party

When a user logs in to a WAM system at the asserting party, an authentication request is initiated. After the WAM system successfully authenticates the user, the identity information is then forwarded to CA SiteMinder® Federation Standalone.

Authentication Initiated at the Relying Party

The relying party can initiate an authentication request, but this scenario is not considered delegated authentication. Delegated authentication occurs only at the asserting party.

A request for a federated resource is made directly to the relying party, who then sends an AuthnRequest to CA SiteMinder® Federation Standalone at the asserting party. CA SiteMinder® Federation Standalone recognizes it as a delegated authentication request and redirects the user to the third-party WAM system at the asserting party. The user logs in to the WAM system, which initiates an authentication request. After the WAM system successfully authenticates the user, the identity information is then forwarded to CA SiteMinder® Federation Standalone.

After the third-party WAM system receives the authentication request, it passes the user identity to CA SiteMinder® Federation Standalone. The method the WAM system uses to pass the user identity depends on whether the delegated authentication method is cookie-based or a query string-based.

How the Third Party WAM Passes the User Identity

The third-party WAM system can use one of two methods to pass a federated user identity to CA SiteMinder® Federation Standalone:

- Using a legacy cookie or open format cookie.

  The open format cookie can be encrypted to ensure the security of the data.

- Using a query string appended to a redirect URL that sends the browser to CA SiteMinder® Federation Standalone.

  The query string is sent in clear text, and it does not produce a FIPS-compliant partnership.

  Important! Do not use the query string method in a production environment. The query string redirection method is only for a testing environment as a proof of concept.

The method a third-party WAM system chooses depends on the configuration it wants to establish for passing a user identity to CA SiteMinder® Federation Standalone.

The methods of passing the user identity are detailed in the following sections.

Cookie Method for Passing User Identity

CA SiteMinder® Federation Standalone can use a legacy or open format cookie to pass a user identity. The cookie contains a user login ID as one of its values.

Note: If you configure delegated authentication for use with the CA SiteMinder® Federation Standalone Agent for Windows Authentication, the Agent requires the use of the open format cookie. However, if the CA SiteMinder® Connector is also configured, the open format cookie option for delegated authentication is not available. The CA SiteMinder® Federation Standalone Windows Agent and the CA SiteMinder® Connector cannot coexist in a deployment.

Authentication can begin at the WAM system or at CA SiteMinder® Federation Standalone. If authentication begins at CA SiteMinder® Federation Standalone, it redirects the user to the WAM system, where the authentication process is the same as if it began at the WAM system.

The delegated authentication process is as follows:

1. An authentication request comes into to the third-party WAM system.
2. The user is authenticated.
3. The third-party WAM system obtains a cookie in one of two ways:
   - The WAM system uses the CA SiteMinder® Federation Standalone SDK to create a legacy cookie or an open format cookie. The SDK creates the cookie and sends it back in a request to the WAM system.

     Note: To create an open format cookie that is FIPS-encrypted, use a CA SiteMinder® Federation Standalone SDK.

     The third-party WAM application must use the same language as the SDK that it is using to create a cookie. If you are using the CA SiteMinder® Federation Standalone Java SDK, the third-party WAM application must be in Java. If you are using the .NET SDK, the third-party WAM application must support .NET.
   - The WAM system uses a manually created open format cookie.

     You can create an open format cookie without using a CA SiteMinder® Federation Standalone SDK. To create the open format cookie cookie manually, use any programming language that supports UTF-8 encoding and any of the following PBE encryption algorithms that CA SiteMinder® Federation Standalone uses for password-based encryption:
     - PBE/SHA1/AES/CBC/PKCS12PBE-1000-128
     - PBE/SHA1/AES/CBC/PKCS12PBE-1000-192
     - PBE/SHA1/AES/CBC/PKCS12PBE-1000-256
     - PBE/SHA256/AES/CBC/PKCS12PBE-1000-128

- PBE/SHA256/AES/CBC/PKCS12PBE-1000-192
- PBE/SHA256/AES/CBC/PKCS12PBE-1000-256
- PBE/SHA1/3DES_EDE/CBC/PKCS12PBE-1000-3
- PBE/SHA256/3DES_EDE/CBC/PKCS12PBE-1000-3

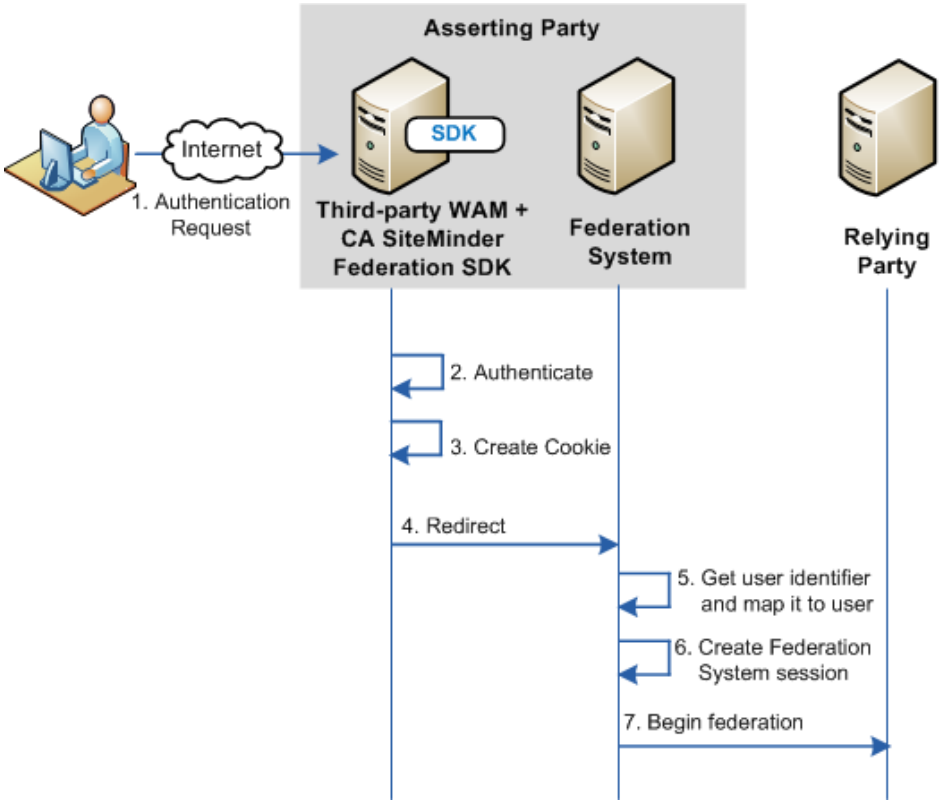
You must also be sure that the open format cookie gets set in the user's browser.

To write a complete cookie, review the details about the contents of the open format cookie (see page 405).

Note: The WAM system and CA SiteMinder® Federation Standalone must be in the same cookie domain.

4. The WAM system redirects the browser to CA SiteMinder® Federation Standalone.
5. CA SiteMinder® Federation Standalone extracts the login ID from the cookie then locates the user in its user directory.
6. CA SiteMinder® Federation Standalone creates a CA SiteMinder® Federation Standalone session.
7. After the session is created, federated communication with the relying party proceeds.

The following picture shows the cookie method when authentication is initiated at the third-party WAM.



Important! To use the legacy cookie or an SDK-created open format cookie, the third party must install a CA SiteMinder® Federation Standalone SDK. The SDK is a separately installed component from CA SiteMinder® Federation Standalone. The installation kit contains the documentation that describes how to use the SDK for delegated authentication.

Query String Method for Passing User Identity

A third-party WAM system can pass a user identity to CA SiteMinder® Federation Standalone by appending a query string on the redirect URL that sends the user from the WAM system to CA SiteMinder® Federation Standalone. For this method to work, the third-party WAM system has to configure a URL that redirects federated users to CA SiteMinder® Federation Standalone after they are authenticated.

Important! Do not use the query string method in a production environment. The query string redirection method is only for a testing environment as a proof of concept.

Notes:

- The query string method does not produce a FIPS-compliant partnership.
- Authentication can also be initiated at CA SiteMinder® Federation Standalone or at the relying party.

If authentication is initiated at the WAM system, the transaction flow for delegated authentication using a query string is as follows:

1. The third-party WAM system receives an authentication request.
2. The user is authenticated.
3. The third-party WAM system constructs a redirect URL and adds the login ID and hashed login ID values to the query string in the format LoginID=*LoginID*&LoginIDHash=*hashed_LoginID.*

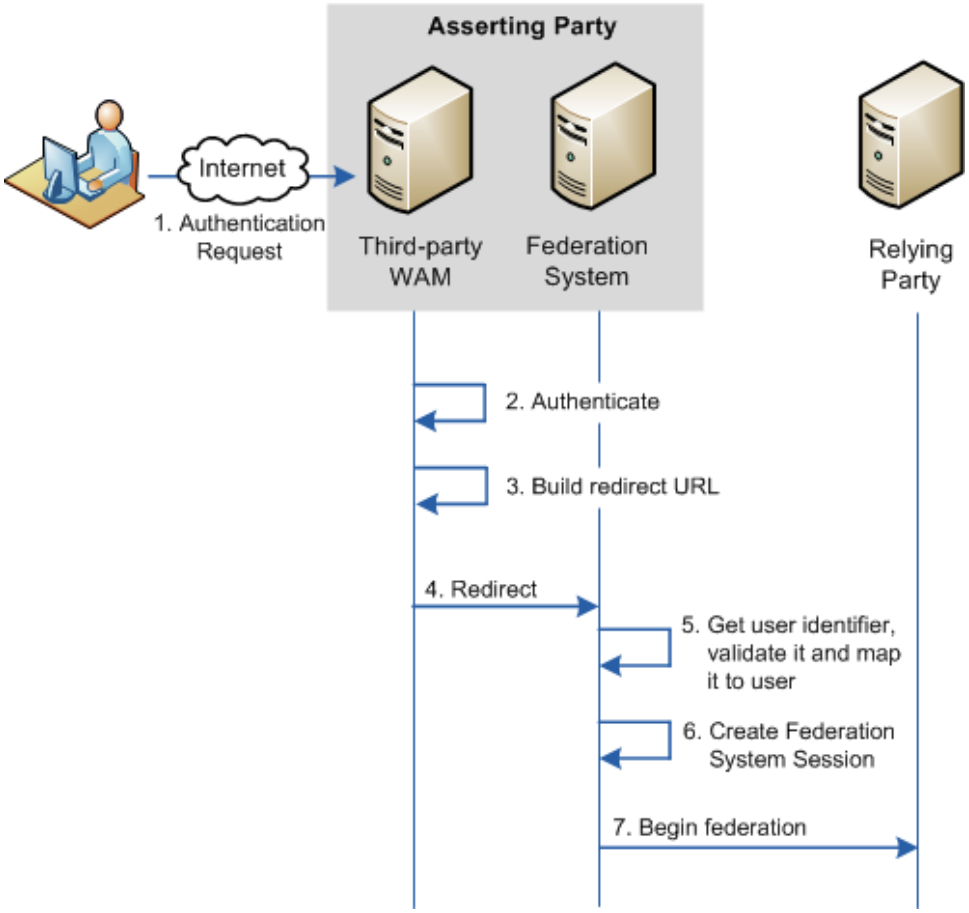
   Important! The LoginID and LoginIDHash parameters are case sensitive. Be sure to include them in the redirect URL as shown in the example.

   The hashing mechanism allows CA SiteMinder® Federation Standalone to verify that the user ID has been received unchanged.

   Example of a Redirect URL

   http://idp1.example.com:9090/affwebservices/public/saml2sso?SPID=FmSP&ProtocolBinding=urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST&LoginID=jdoe&LoginIDHash=454d3bd5cb839168eeffcf060ae0b9c28ed6eec0
4. The WAM system redirects the browser to CA SiteMinder® Federation Standalone.
5. CA SiteMinder® Federation Standalone extracts the login ID and hashed login ID from the URL, validates the identifier using the hashed value, and locates the user in its user directory.
6. CA SiteMinder® Federation Standalone creates a user session.
7. After the session is created, federated communication with the relying party proceeds.

The following graphic illustrates the query string method when authentication is initiated at the asserting party.



Delegated Authentication Configuration

Delegated authentication is configured at the asserting party, where an assertion is generated based on an authenticated user identity.

To configure delegated authentication

1. Determine which method (cookie or query string) the third-party WAM uses to pass the user identity.

   Note: The query string does not produce a FIPS-compliant partnership.

2. Go to the appropriate step in the partnership wizard to set up delegated authentication.

Important! To use the SDK-created open-format cookie, the third party must install a CA SiteMinder® Federation Standalone SDK. The SDK is a separately installed component. The installation kit contains the documentation that describes how to use the SDK for delegated authentication.

More information

Deployment Settings (see page 346)

Cookie Delegated Authentication Sample Setup

The following sample configuration is from the perspective of a SAML 2.0 IdP > SP partnership. The delegated authentication settings are on the SSO and SLO step of the partnership wizard.

This sample configuration reflects a SAML 2.0 configuration. The Identity Provider is http://idp1.xyz.com and the third-party WAM system is http://wamservice.xyz.com.

To configure cookie delegated authentication

1. Create a partnership or edit an existing one.

   Note: To edit a partnership, deactivate it first.

2. Navigate to the SSO and SLO step in the Partnership wizard.

3. In the Authentication section, set the fields as follows:

   Authentication Mode

   Delegated

   Delegated Authentication Type

   Open format cookie

   For use with a web access management application. You can use a CA SiteMinder® Federation Standalone SDK to create a Java or .NET application. Alternatively, you can use an application written in another language, provided you build the open-format cookie manually.

   If you require FIPS 140-2 encryption, create the open-format cookie using the CA SiteMinder® Federation Standalone Java or .NET SDK.

   Delegated Authentication URL

   http://wamservice.xyz.com

   The URL of the third-party WAM system that authenticates users and uses a CA SiteMinder® Federation Standalone SDK to create the cookie.

   Authentication Class

   Enter the authentication method that is used at the third party. For example:

   urn:oasis:names:tc:SAML:2.0:ac:classes:Kerberos

4. Communicate all the open-format cookie settings to the third-party WAM system.

   CA SiteMinder® uses these values in the creation of the cookie.

5. Continue with partnership configuration.

Query String Delegated Authentication Sample Setup

The following sample configuration is from the perspective of a SAML 2.0 IdP > SP partnership. The delegated authentication settings are on the SSO and SLO step of the partnership wizard.

Note: The query string method does not produce a FIPS-compliant partnership.

This sample configuration reflects a SAML 2.0 configuration. The Identity Provider is http://idp1.xyz.com and the third-party WAM system is http://wamservice.xyz.com.

Important! Do not use the query string method in a production environment. The query string redirection method is only for a testing environment as a proof of concept.

To configure query string delegated authentication

1. Create a partnership or edit an existing one.

   Note: To edit a partnership, deactivate it first.

2. Navigate to the appropriate step in the partnership wizard.

3. In the Authentication section, set the fields as follows:

   Authentication Mode

   Delegated

   Delegated Authentication Type

   Query String

   Delegated Authentication URL

   http://wamservice.xyz.com

   The URL of the third-party WAM system that authenticates users and constructs the redirect URL back to CA SiteMinder® with the query parameters.

   Hash Secret

   FederatedAuth1

   The third-party WAM system uses this secret to hash the login ID.

   Confirm Hash Secret

   FederatedAuth1

   Authentication Class

   Enter the authentication method that is used at the third party. For example:

   urn:oasis:names:tc:SAML:2.0:ac:classes:Kerberos

4. Continue with partnership configuration.

Third-party WAM Configuration for Cookie Delegated Authentication

For delegated authentication to succeed, the third-party WAM must adjust its federated application, as follows:

- To communicate the authenticated user login ID through a cookie, the third-party WAM system must generate a cookie.
  - For Java applications, the WAM can use a CA SiteMinder® Federation Standalone Java SDK to create a legacy cookie or an open format cookie.
  - For .NET applications, the WAM can use a CA SiteMinder® Federation Standalone .NET SDK to create an open format cookie.
  - For languages other than Java and .NET, the WAM can create an open format cookie manually.

  For details on implementing the necessary class and methods, see the *CA SiteMinder® Federation Standalone Java SDK Guide* or the *CA SiteMinder® Federation Standalone .NET SDK Guide.* Each guide is installed with the SDK. If you create an open format cookie manually, review the details about the required contents of the cookie (see page 405).
- The third party must know the values of the following Administrative UI settings Cookie Zone and Encryption Password parameters configured at the CA SiteMinder® Federation Standalone asserting party:
  - Global Cookie Zone
  - Encryption Password
  - Open-format Cookie Name
  - Open-format Cookie Encryption Transformation

  These values are used in the creation of the cookie.
- The third-party WAM system must create a redirect URL that sends the user back to CA SiteMinder® Federation Standalone. This URL has to send the user back to the CA SiteMinder® Federation Standalone Single Sign-on service. The CA SiteMinder® Federation Standalone Administrator has to communicate the Single Sign-on service to the third party in an out-of-band communication.

  Important! After the third-party WAM system receives an authentication request from CA SiteMinder® Federation Standalone, it must capture and resend any existing query string it receives as part of the incoming authentication request. The incoming request can have CA SiteMinder® Federation Standalone request information within the query string and must be passed along unchanged.

  Note: To pass the cookie, the third-party WAM system must be in the same cookie domain as CA SiteMinder® Federation Standalone at the asserting party.

Third-party WAM Configuration for Query String Delegated Authentication

A third-party WAM system and CA SiteMinder® Federation Standalone at the asserting party communicate the login ID in a query string. The WAM system must add the following two attributes to the query string in the redirect URL:

LoginID

Specifies the value used to identify the user to the third-party WAM system.

LoginIDHash

A hash of the LoginID.

To generate the LoginIDHash value, the LoginID is prepended to a Hash Secret and the entire value is then run through a SHA-1 hashing algorithm. The Hash Secret is specified in the CA SiteMinder® Federation Standalone configuration at the asserting party.

When CA SiteMinder® Federation Standalone retrieves the credentials from the query string, it also combines these values and hashes them. If the hashes are equal, CA SiteMinder® Federation Standalone considers the login ID to be valid and continues with the federation request.

Important! The LoginID and LoginIDHash parameters are case sensitive.

The third-party WAM system must configure its federated application to construct a redirect URL that sends the user back to the CA SiteMinder® Federation Standalone Single Sign-on service. Therefore, the CA SiteMinder® Federation Standalone Administrator has to communicate the Single Sign-on service to the third party in an out-of-band communication.

Important! After the third-party WAM system receives an authentication request from CA SiteMinder® Federation Standalone, it must remember to capture and resend any existing query string it receives as part of the incoming authentication request. If the incoming request has CA SiteMinder® Federation Standalone request information within the query string it must be passed along unchanged.

The syntax of the query string is as follows:

?*existing_query_string*&LoginID=*LoginID*&LoginIDHash=*hashed_LoginID*

Example

```
https://johndoe3227.b.com/affwebservices/public/saml2sso?SPID=sp1&
LoginID=user1&LoginIDHash=de164152ed6e8e9a7f760e47d135ecf0c98a
3e4e&ProtocolBinding=urn:oasis:names:tc:SAML:2.0:bindings:HTTP-Artifact
```

Chapter 15: URLs to Initiate Single Sign-on

This section contains the following topics:

Links to Servlets which Initiate Single Sign-on (see page 239)
Producer-initiated SSO (SAML 1.1) (see page 239)
IdP-initiated SSO (SAML 2.0 Artifact or POST) (see page 241)
SP-initiated SSO (SAML 2.0) (see page 244)
IP-initiated Single Sign-on (WSFED) (see page 248)
RP-initiated Single Sign-on (WSFED) (see page 249)

Links to Servlets which Initiate Single Sign-on

When designing a site for federated content, that site includes a page with specific links to trigger single sign-on. These links are URLs to servlets for the Single Sign-on service or the AuthnRequest Service.

To initiate single sign-on, the user can begin at the asserting or relying party. Configure the appropriate links at each site to initiate single sign-on operation.

Producer-initiated SSO (SAML 1.1)

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. The URL makes a request to the producer-side web Agent before the user is redirected to the consumer site.

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

```
http://producer_host:port/affwebservices/public/intersitetransfer?
CONSUMERID=consumer_entity_ID&TARGET=http://consumer_site/target_url
```

The variables and query parameters in the previous intersite transfer URL are as follows:

producer_host:port

Specifies the server and port number where the user is authenticated.

CONSUMERID

(Required) Identifies the consumer. On the producer side, the producer-to-consumer partnership has a name, and the remote consumer entity has an ID. The CONSUMERID is the entity ID of the remote consumer.

You can use the parameter NAME in place of CONSUMERID, but not both.

If you use NAME, specify the name of the producer-to-consumer partnership as defined at the producer.

consumer_entity_ID

Identifies the consumer site the user wants to visit from the producer site.

TARGET

(Optional) Identifies the requested target resource at the consumer.

The TARGET parameter is optional. You are required to define the target; however, you can define it in the consumer-side partnership instead of the intersite transfer URL. The target is defined in the Application Integration step of the Partnership wizard. Be sure to define the target in the URL or in the partnership.

consumer_site

Specifies the server at the consumer site.

target_url

Indicates the target application at the consumer site.

Note: Query parameters for the SAML Artifact binding must use HTTP-encoding.

Example of an intersite transfer URL for the Artifact and POST profile:

```
http://www.smartway.com/affwebservices/public/intersitetransfer?
CONSUMERID=ahealthco&TARGET=http://www.ahealthco.com:85/
smartway/index.jsp
```

IdP-initiated SSO (SAML 2.0 Artifact or POST)

If a user visits a CA SiteMinder® Federation Standalone Identity Provider before going to the Service Provider, an unsolicited response at the Identity Provider must be initiated. To initiate an unsolicited response, create a hard-coded link that generates an HTTP Get request that CA SiteMinder® Federation Standalone accepts. This HTTP Get request must contain a query parameter that provides the Service Provider ID. The Identity Provider must generate the SAML assertion response. A user clicks this link to initiate the unsolicited response.

Note: This information applies to Artifact or POST bindings.

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*&RelayState=*target_URL*

idp_server:port

Identifies the web server and port hosting CA SiteMinder® Federation Standalone.

SP_ID

Specifies the Entity ID of the Service Provider defined in the partnership.

URI_for_binding

Identifies the URI of the POST or Artifact binding for the ProtocolBinding element. The SAML 2.0 specification defines this URI.

- The URI for the artifact binding, as specified by the SAML 2.0 specification is:

  urn:oasis:names:tc:SAML:2.0:bindings:HTTP-Artifact

- The URI for the POST binding, as specified by 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: A binding must also be enabled for the partnership for the request to work.

target_URL

Specifies the URL of the federation resource target at the Service Provider.

Note the following:

- If you do not include the ProtocolBinding query in the link, use the one binding configured in the Service Provider properties
- When Artifact and POST are enabled in the Service Provider properties, POST is the default. Therefore, if you only want to use Artifact binding, include the ProtocolBinding query parameter in the link.

Important! If you configure indexed endpoint support for Assertion Consumer Services, the value of the ProtocolBinding query parameter overrides the binding for the Assertion Consumer Service.

Unsolicited Response Query Parameters Used by the IdP

An unsolicited response that initiates single sign-on from the IdP can include the following query parameters:

SPID

(Required) Specifies the ID of the Service Provider where the Identity Provider sends the unsolicited response.

ProtocolBinding

Specifies the ProtocolBinding element in the unsolicited response. This element specifies the protocol for sending the assertion response to the Service Provider. If the Service Provider is not configured to support the specified protocol binding, the request fails.

RelayState

Indicates the URL of the target resource at the Service Provider. By including this query parameter, it tells the IdP to redirect the user the appropriate resource at the Service Provider. This query parameter can be used in place of specifying a target URL when configuring single sign-on.

Required Use of the ProtocolBinding Query Parameter

The ProtocolBinding query parameter is required *only* if the artifact and POST binding are enabled for the Service Provider properties. In addiiton, the user wants to only use artifact binding.

- The URI for the artifact binding is:

  urn:oasis:names:tc:SAML:2.0:bindings:HTTP-Artifact

- The URI for the POST binding 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: HTTP coding the query parameters is not necessary.

Optional Use of the ProtocolBinding Query Parameter

When you *do not* use the ProtocolBinding query parameter, the following information applies:

- If only one binding is enabled for the Service Provider and the ProtocolBinding is not specified in the unsolicited response, the enabled binding is used.
- If both bindings are enabled for the Service Provider and the ProtocolBinding is not specified in the unsolicited response, the POST binding is the default.

Example: Unsolicited Response without ProtocolBinding

The link redirects the user to the Single Sign-on service. Included in this link is the Service Provider identity, which the SPID query parameter specifies. The ProtocolBinding query parameter is not present. After the user clicks this hard-coded link, they are redirected to the Single Sign-on service.

http://fedsrv.fedsite.com:82/affwebservices/public/saml2sso?
SPID=http%3A%2F%2Ffedsrv.acme.com%2Fsmidp2for90

Example: Unsolicited Response with ProtocolBinding

The link redirects the user to the Single Sign-on service. Included in this link is the Service Provider identity, which the SPID query parameter specifies and the artifact binding is being used. After the user clicks this hard-coded link, they are redirected to 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

ForceAuthn and IsPassive Processing at the IdP

If single sign-on is initiated by a Service Provider, that Service Provider may include a ForceAuthn or IsPassive query parameter in an AuthnRequest message.

Note: CA SiteMinder® Federation Standalone Identity Providers do not support the IsPassive query parameter; however, the IsPassive parameter may be included in an AuthnRequest message sent by a third-party Service Provider.

When a Service Provider includes ForceAuthn or IsPassive in the AuthnRequest, a CA SiteMinder® Federation Standalone Identity Provider handles these query parameters as follows:

ForceAuthn Handling

When a Service Provider includes ForceAuthn=True in the AuthnRequest message, a CA SiteMinder® Federation Standalone Identity Provider challenges the user for their credentials, regardless of whether or not a CA SiteMinder® Federation Standalone session exists. If the user successfully authenticates, a session is established.

IsPassive Handling

When a Service Provider includes IsPassive in the AuthnRequest and it cannot be honored by the Identity Provider, one of the following SAML responses is sent back to the Service Provider:

- If IsPassive=True in the AuthnRequest message and there is no CA SiteMinder® Federation Standalone session, a CA SiteMinder® Federation Standalone Identity Provider returns a SAML response that includes an error message because CA SiteMinder® Federation Standalone requires a session.
- If IsPassive=True in the AuthnRequest message and there is a CA SiteMinder® Federation Standalone session, the CA SiteMinder® Federation Standalone Identity Provider returns the assertion.
- If IsPassive and ForceAuthn are in the AuthnRequest message and both are set to True, the CA SiteMinder® Federation Standalone Identity Provider returns an error because this is an invalid request. IsPassive and ForceAuthn are mutually exclusive.

SP-initiated SSO (SAML 2.0)

SP-initiated SSO requires that you have an HTML page at the Service Provider containing hard-coded links to the AuthnRequest service at the Service Provider. The links redirect the user to the Identity Provider to be authenticated and determining what is included in the AuthnRequest itself.

This information applies to Artifact or POST bindings.

The hard-coded link that the user selects must contain specific query parameters, which are used in an HTTP GET request to the AuthnRequest service.

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*&
RelayState=*target_URL*

sp_server:port

Specifies the server and port number at the Service Provider that is hosting CA SiteMinder® Federation Standalone.

IdP_ID

Specifies the identity that is assigned to the Identity Provider.

URI_of_binding

Identifies the URI of the POST or Artifact binding for the ProtocolBinding element. The SAML 2.0 specification defines this URI.

- The URI for the artifact binding is:

  urn:oasis:names:tc:SAML:2.0:bindings:HTTP-Artifact

- The URI for the POST binding 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.

Also, enable a binding for the partnership for the request to work.

target_URL

Specifies the URL of the federation target at the Service Provider.

Note the following information:

- If you do not include the ProtocolBinding query parameter in the AuthnRequest link, the default binding is the one defined for the partnership. If you have both bindings defined in the partnership, then no binding is passed in the AuthnRequest. As a result, the default binding at the Identity Provider is used.
- If the artifact and POST bindings are enabled for the partnership but you only want to use artifact binding, include the ProtocolBinding query parameter in the link.

AuthnRequest Query Parameters Used by an SP

The query parameters a CA SiteMinder® Federation Standalone SP can use in the links to the AuthnRequest Service are as follows:

ProviderID (required)

Entity ID of the Identity Provider where thethe AuthnRequest Service.sends AuthnRequest message.

ProtocolBinding

Specifies the ProtocolBinding element in the AuthnRequest message. This element specifies the protocol used to return the SAML response from the Identity Provider. 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.

ForceAuthn

Instructs the Identity Provider that it must authenticate a user directly instead of relying on an existing security context. Use this query parameter when the Identity Provider is using CA SiteMinder® Federation Standalone, not if it is using third-party federation software.

- If the SP sets ForceAuthn=True in the AuthnRequest message, and a session exists for a particular user, the Identity Provider challenges the user for credentials. If the user successfully authenticates, the IdP sends the identity information from the existing session in the assertion and discards the session generated for the authentication.
- If the SP sets ForceAuthn=True in the AuthnRequest message and there is no session, the IdP challenges the user for credentials. If the user successfully authenticates, a session is established.

Example

http://sp1.demo.com:81/affwebservices/public/saml2authnrequest?ProviderID=idp1.example.com&ForceAuthn=yes

AssertionConsumerServiceIndex

Specifies the index of the endpoint acting as the Assertion Consumer Service. It tells the Identity Provider where to send the assertion response.

If you use this parameter in the AuthnRequest, do not include the ProtocolBinding parameter also because they are mutually exclusive. The Assertion Consumer Service has its own protocol binding, which could conflict with the ProtocolBinding parameter.

RelayState

Indicates the URL of the target resource at the Service Provider. By including this query parameter, it tells the Service Provider where to send the user. Otherwise, the default target defined for the partnership is used.

Required Use of the ProtocolBinding Query Parameter

The ProtocolBinding parameter is required if artifact and POST binding are enabled for the partnership, and if the user wants to use only the artifact binding.

- The URI for the artifact binding, as specified by the SAML 2.0 specification is:

  urn:oasis:names:tc:SAML:2.0:bindings:HTTP-Artifact

- The URI for the POST binding as specified by 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.

Optional Use of ProtocolBinding

When you *do not* use the ProtocolBinding query parameter, the following applies:

- If only one binding is enabled for the partnership and the ProtocolBinding query parameter is not specified, the enabled binding for the partnership is used.
- If both bindings are enabled and the ProtocolBinding query parameter is not specified, POST binding is used as the default.

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

Example: AuthnRequest Link without the ProtocolBinding Query Parameter

This sample link goes to the AuthnRequest service. It 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

After a user clicks the link at the Service Provider, CA SiteMinder® Federation Standalone passes a request for an AuthnRequest message.

Example: AuthnRequest Link with the ProtocolBinding Query Parameter the

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

After a user clicks the link at the Service Provider, CA SiteMinder® Federation Standalone passes a request for an AuthnRequest message.

IP-initiated Single Sign-on (WSFED)

A user can visit the Identity Provider (IP) before going to the Resource Partner (RP). If the user visits the Identity Provider first, a link must generate an HTTP Get request. The hard-coded link points to the passive requester service at the IP. The request contains the RP Provider ID and optionally other parameters.

The syntax for the link is:

https://i*p_server:port*/affwebservices/public/wsfedsso?wa=wsignin1.0&wtrealm=*rp_id*

ip_server:port

Specifies the server and port number of the system at the Identity 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

The ID of the RP.

RP-initiated Single Sign-on (WSFED)

When a user starts at the RP to initiate single sign-on, typically the user selects from a list of IPs. The site selection page is in an unprotected realm.

The link on the site selection page points to the passive requester service at an IP. After the link is selected, the RP redirects the user to the IP to get the assertion.

Chapter 16: Logging Out of User Sessions

This section contains the following topics:

Single Logout (SAML 2.0) (see page 251)
Sign-Out Overview (WS-Federation) (see page 257)
Local Logout at the SP (SAML 2.0) (see page 258)

Single Logout (SAML 2.0)

Single logout (SLO) results in the simultaneous termination of all user sessions for the browser that initiated the logout. Closing all user sessions prevents unauthorized users from gaining access to resources at the SPs.

The single logout binding determines what is sent with a single logout message and how each received message is handled.

Two bindings are available for single logout operation:

HTTP-Redirect

HTTP-Redirect binding relies on a browser to conduct each logout transaction. The single logout message is always a GET request. The browser is involved in every request and response. The involvement of the browser means that HTTP-redirect binding provides browser session data, which the SOAP binding does not.

A disadvantage of HTTP-Redirect binding is that the data in the message is limited to what you can send on the query string. Also, HTTP-Redirect binding is an asynchronous process so timeouts are unlikely. However, if a redirect fails, that failure stops the entire single logout chain.

SOAP

SOAP binding uses POST requests to conduct single logout transactions. POST requests let you send more data than the HTTP-Redirect binding. SOAP also enables you to do more in the way of encryption and other features.

SOAP is a synchronous process. The IdP has more control and can prevent a problem at a single SP from interfering with the whole process. SOAP communication takes place over a back channel. One logout failure does not have to stop the IdP from attempting to log out from the rest of the SPs.

SOAP relies on a back channel connection, so after the initial single logout call and response a browser is not involved. The the SOAP binding does not clean up cookies at the remote entity as part of the logout process. Cookies are cleaned up only at the local entity. If deleting cookies is required, use HTTP-Redirect binding.

Managing Single Logout Across a Network Using HTTP-Redirect and SOAP

Your network could have some sites that support the HTTP-Redirect binding and others that support the SOAP binding. The IdP has to manage multiple bindings, but the SP sends or receives only one logout request.

The following sections provide configuration guidelines to handle a mixed-binding environment.

SLO Configuration if CA SiteMinder® Federation Standalone is at the IdP

If CA SiteMinder® Federation Standalone is at the IdP, configure the partnership to include an HTTP Redirect-based SLO Service URL and a SOAP-based SLO Service URL.

CA SiteMinder® Federation Standalone at the IdP inspects the configuration for each SP in a session and handles all SOAP-enabled logouts first. HTTP-Redirect logouts for SPs that do not support SOAP follow.

SLO Configuration if CA SiteMinder® Federation Standalone is at the SP

If CA SiteMinder® Federation Standalone is at the SP and the SP initiates single logout, we recommend the HTTP-Redirect binding to initiate the logout. Other SPs for the user session possibly do no support SOAP.

HTTP-Redirect relies on a browser session to handle all redirections. For this reason, it sends the necessary data that the IdP must have to logout SPs that only support HTTP-Redirect. If the initiating SP starts the process with HTTP-Redirect, the IdP can use SOAP with all SPs that support it. Switch to HTTP-Redirect binding for the remaining SPs.

If you initiate single logout with the SOAP binding, the browser session data is not present.

To help ensure an SP-initiated logout uses HTTP-Redirect, embed an HTTP-Redirect link that points to the SP' local servlet in a page or application. For CA SiteMinder® Federation Standalone, that link is:

http://*sp_host:port*/affwebservices/public/saml2slo.

This embedded link causes CA SiteMinder® Federation Standalone to generate a SAML <LogoutRequest> message that it sends to the SLO service at the IdP. When a user logs out, the logout at the SP is performed first and then the logout request is sent to the IdP. The IdP then completes the logout process with all the other SPs involved in the user session.

Understanding Skew Time for SLO Request Validity

Two values are relevant when calculating how long the logout request is valid. These values are 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, CA SiteMinder® Federation Standalone takes its system time. The resulting time becomes the IssueInstant set in the request message. To determine when the logout request expires, CA SiteMinder® Federation Standalone takes its current system time and adds the Skew Time plus the SLO Validity Duration. The resulting time becomes the NotOnOrAfter value.

Note: Times are relative to GMT.

For example, a log out request is generated at the asserting party 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.

Configure Single Logout

Be aware of the following information when configuring single logout:

- If a partner receives a SAML <LogoutRequest> message using HTTP-Redirect, the response back to the sending party must use HTTP-Redirect.
- If a partner receives SAML <LogoutRequest> message using SOAP, the response back to the sending party must be over SOAP.
- If a partner receives an SLO request over a binding it does not support, single logout fails.
- If a single logout user session includes partners that use HTTP-Redirect and SOAP, configure CA SiteMinder® Federation Standalone to support both bindings. When the IdP proceeds with the logout, it logs out all SPs using SOAP then logs out all SPs using HTTP-Redirect binding.
- If a CA SiteMinder® Federation Standalone SP initiates single logout, we recommend starting with HTTP-Redirect binding, even if the SP supports SOAP.

  Review configuration guidelines (see page 252) for managing single logout in an environment that supports SOAP and HTTP-Redirect.

To configure single logout at either side of a partnership

Note: The SLO configuration settings are the same at the IdP and SP.

1. Begin at the SSO and SLO step of the Partnership wizard.
2. In the SLO section, select one or both SLO bindings.

   The SLO binding enables single logout and indicates the binding in use at the local entity. The SLO binding also indicates which binding the local entity accepts when it receives a single logout request.

   If you select SOAP, you can encrypt the Name ID in the SOAP message. The setting for this option is in the Signature and Encryption step of the Partnership wizard.

   If you select SOAP as the binding, the Incoming and Outgoing Configuration for the Back Channel becomes active. SLO requests and responses are sent across a back channel. Each local partner can secure the back channel by requiring the remote partner to authenticate.

   More information can be found about the back channel settings for SLO.
3. Configure any of the additional SLO settings:
   - SLO Confirm URL
   - Validity Duration
   - Relay State overrides SLO Confirm URL

   Note: Click Help for a description of fields, controls, and their respective requirements.
4. Complete the table for the SLO Service URLs. You must have at least one entry.

   The SLO Service URL initiates single logout, which then triggers CA SiteMinder® Federation Standalone to generate a SAML <LogoutRequest> message. In addition, the SLO Service URL tells CA SiteMinder® Federation Standalone where to send the logout request message.

   Specify a SLO service URL for each supported SLO binding, as follows:
   - HTTP-Redirect enabled—select one URL with HTTP-Redirect as the binding.
   - SOAP enabled—select one URL with SOAP as the binding.
   - Redirect and SOAP enabled—select two URLs, one set to HTTP-Redirect and one set to SOAP.

   Note: The Response Location URL field is optional.

   Click Add Row to add more entries to the table. Values defined for the selected remote entity are already entered in the table.

Single logout is configured after these steps are complete.

Back Channel Configuration for Single Logout

Single logout enabled with the SOAP binding sends logout requests and responses across a back channel. You can require an entity to authenticate to access the back channel. The back channel can also be secured using SSL, though SSL is not required.

Securing the back channel using SSL involves:

- Enabling SSL.

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

- Configure an incoming and outgoing back channel for the single logout communication exchange. The local entity has to be able to send messages over the outgoing channel and receive messages over the incoming channel.

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

- Choosing the type of authentication required for the remote entity to gain access across the protected back channel. The authentication method applies per channel (incoming or outgoing).

  The options for back channel authentication are:

  Basic

  Indicates that a Basic authentication scheme is protecting the back channel.

  Note: If SSL is enabled for the back channel connection, Basic authentication can still be selected.

  Client Cert

  Indicates that SSL with an X.509 client certificate protects the asserting party back channel.

  If you select Client Cert as the authentication method, 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).

NoAuth

Indicates that the relying party is not required to supply credentials. The back channel is not secured. You can still enable SSL with this option. The back channel traffic is encrypted but no credentials are exchanged between parties.

Use the NoAuth option for testing purposes but not for production, unless CA SiteMinder® Federation Standalone is configured for SSL-enabled failover and sits behind a proxy server. In this case, if client certificate authentication is used to protect the back channel, the proxy server handles the authentication because it has the server certificate. In that case, all IdP->SP partnerships can use NoAuth as the authentication type.

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

To secure the back channel for single logout

1. Begin at the Back Channel group box in the SSO and SLO step of the Partnership wizard.
2. Select SOAP in the SLO group box. The Authentication Method field becomes active.
3. Select the type of authentication method for the incoming and outgoing back channel. Additional fields to configure are displayed for Basic and Client Cert methods.

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

   If you select No Auth as the authentication method, no additional steps are required.
4. Depending on the authentication method you select, several additional fields are displayed for you to configure.

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

After entering values for all the necessary fields, the back channel configuration is complete.

More information:

SSL Administration for the Apache Web Server and the UI (see page 359)

Sign-Out Overview (WS-Federation)

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 partner 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 WSFED Sign-Out

Requirements to configure sign-out:

- To enable sign-out at the Identity Provider, enable the session store using the Policy Server Management Console.

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

- Configure persistent sessions for the realm with the protected resources at the Resource Partner.

  For information about realms, see the *Policy Server Configuration Guide*.

Follow these steps:

1. Log in to the Administrative UI.
2. Select the WS-Federation partnership that you want to modify.
3. Navigate to the Single Sign-on and Sign-Out step of the partnership wizard.

4. In the Sign-Out section, set the following fields:
   - Enable Sign Out
   - Sign-Out Confirm URL (IP only)
   - Sign-Out URL

   The URLs must each have an entry that starts with https:// or http://.

   Note: Click Help for a description of fields, controls, and their respective requirements.
5. Navigate to the Confirm step and click Finish to save your changes.

Sign-out is configured.

Local Logout at the SP (SAML 2.0)

CA SiteMinder® as an SP supports local logout for stand-alone applications. Local logout enables a user to be logged out at the local SP-side application. The session at the SP is removed, but no communication with the IdP or other SPs is involved. Sessions at the IdP and other SPs remain active.

If you include a logout link in an application at the SP, the SP sends a logout request to the local single logout service. The SP logs out the user upon receiving the request. The application at the SP is responsible for sending a confirmation message that the logout is successful.

CA SiteMinder® provides local logout using a query parameter named **localLogout**. To use this parameter, your application can have a page, such as the following example:

```
You have completed your registration with demoapp.
To end your session securely, select LOGOUT.
```

The following sample string represents the link for the LOGOUT button:

```
<http://sp1server.demo.com:8080/affwebservices/public/saml2slo?Loc
alLogout=true
```

Chapter 17: Authentication Context Processing (SAML 2.0)

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

Requesting the Authentication Context

A CA SiteMinder® Service Provider requests the authentication context by including the <RequestedAuthnContext> element in the authentication request to the Identity Provider. Inclusion of this element is based on a configuration setting in the SP->Identity Provider partnership.

Obtaining the Authentication Context

A CA SiteMinder® Identity Provider obtains the authentication context for a user in one of two ways:

- You specify a static AuthnContext URI in the IdP->SP partnership configuration.
- If you are communicating with a CA SiteMinder® Service Provider that does not support AuthnContext requests, manually enter a URI.
- The AuthnContext URI is determined dynamically using a configured authentication context template.

  CA SiteMinder® maps the authentication context URIs to CA SiteMinder® authentication levels. CA SiteMinder® authentication levels indicate the strength of an authentication context for an established user session. The levels enable the authentication context to be derived from the user session at the Identity Provider.

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

This section contains the following topics:

Determine how a User Authenticated at an Identity Provider (see page 260)
Determine Authentication Context and Strength Levels with your Partner (see page 264)
Configure an Authentication Context Template (see page 264)
Configure Authentication Context Processing at the IdP (see page 268)
Configure Authentication Context Requests at the SP (see page 270)

Determine how a User Authenticated at an Identity Provider

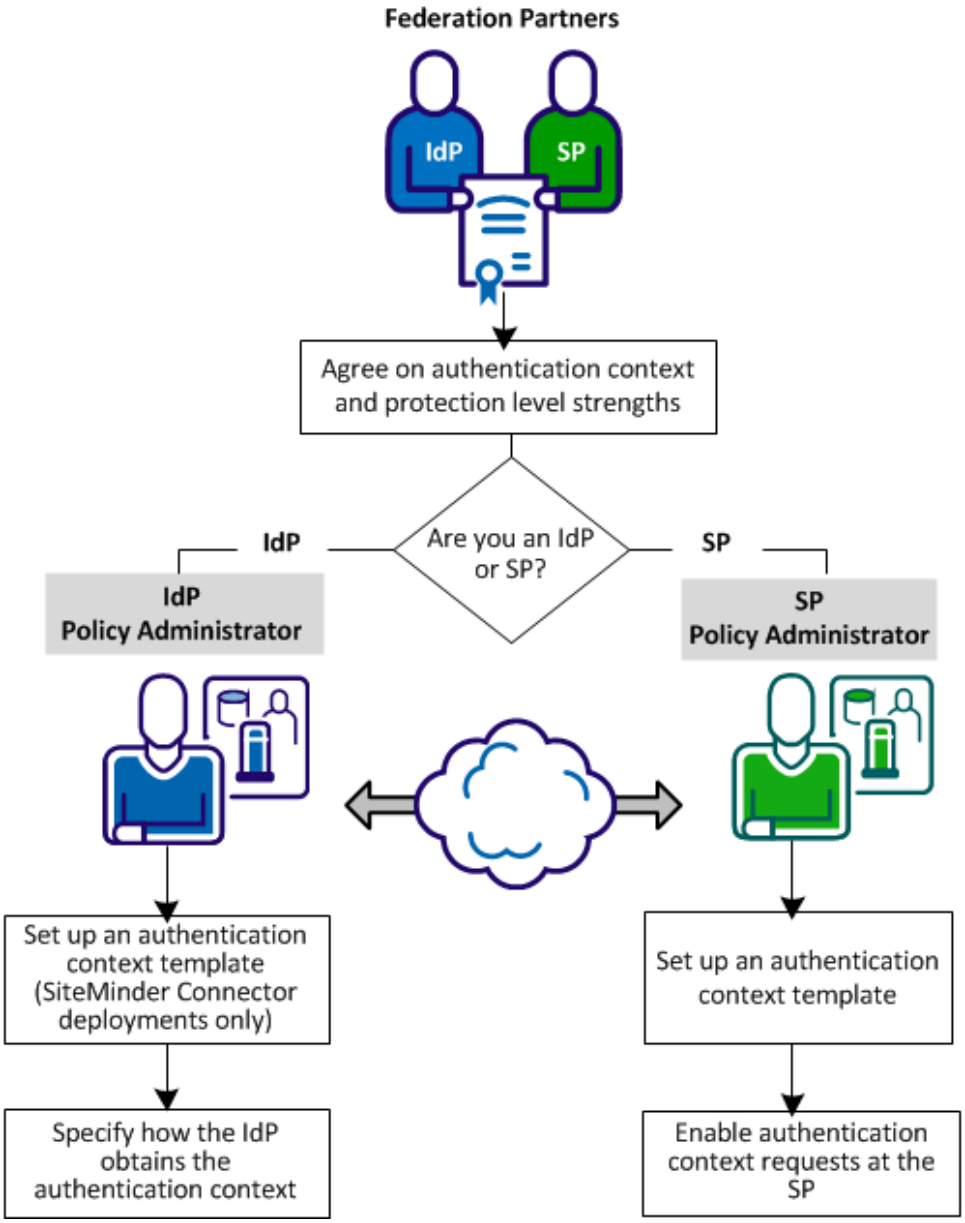
The *authentication context* indicates how a user authenticated at an Identity Provider. The Identity Provider includes the authentication context in an assertion at the request of a Service Provider or based on configuration at the Identity Provider. A Service Provider can require information about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

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

Verify that the policy administrators meet the following minimum knowledge requirements:

- Familiarity with SAML 2.0 specifications and standards related to authentication context processing.
- Understand CA SiteMinder® Federation Standalone configuration objects.
- Know how to access and use the CA SiteMinder® Federation Standalone Administrative UI.

The following figure shows the configuration process for each partner. CA SiteMinder® Federation Standalone does not have to be installed at each site.



Complete the following steps to configure authentication context processing:

1. Agree on authentication context and protection level strengths (see page 264).
2. Set up an authentication context template (see page 265).
3. Complete the task for your site:
   - Specify how the IdP obtains the authentication context (IdP-side) (see page 268)
   - Enable authentication context requests at the SP (SP-side) (see page 271)

Authentication Context Processing for IdP-initiated SSO

When single sign-on is initiated at the IdP, authentication context processing follows these steps:

1. A user request triggers single sign-on at the IdP.
2. The user is authenticated and a user session is generated. Associated with the session is a protection level that is configured with the authentication scheme.
3. Depending on the authentication context configuration at the IdP, *one* of the following conditions occur:
   - Automatic detection occurs—only available if the SiteMinder Connector is enabled for the IdP-to-SP partnership.

     Based on a configured authentication context template, the AuthnContext class is mapped to the protection level for the session.
   - Predefined authentication class is used.

     The hard-coded URI you specify is added to the assertion.
4. The IdP generates the assertion and adds the authentication context to it. The assertion is then sent to the SP.
5. At the SP, another comparison is made between the authentication context class from the assertion and the one configured at the SP. If this comparison is successful, the authentication transaction is complete.

Authentication Context Processing for SP-Initiated SSO

When single sign-on is initiated at the SP, authentication context processing follows these steps:

1. The SP sends an authentication request with the <RequestedAuthnContext> element and a comparison operator. The element is included based on a setting in the configuration of the SP-> IdP partnership.

2. When the IdP receives the request, the IdP authenticates the user and a user session is generated. Associated with the session is a protection level for the authentication scheme.
3. Depending on the authentication context configuration at the IdP, *one* of the following conditions occur:
   - Automatic detection occurs

     Based on a configured authentication context template, the AuthnContext class is mapped to the protection level for the session.
   - Predefined authentication class is used

     The hard-coded URI you specify is added to the assertion.
4. The IdP compares the AuthnContext against the authentication class for the user session. The comparison is based on the comparison operator that is sent with the request. See the table that follows this procedure for examples of how each comparison operator affects processing.

   If the SP includes multiple authentication context URIs in the request, the classes are compared one-by-one in sequential order against the context for the session. At the first successful comparison, the IdP adds the session authentication context to the assertion.
5. If the comparison is successful, then the authentication context is added to the assertion sent to the SP.

   If the comparison is not successful, the transaction is terminated with a "noauthncontext" status response.
6. At the SP, a second comparison takes place between the authentication context from the assertion and the one configured at the SP. If this comparison is successful, the authentication transaction is complete.

The following table shows examples of how an authentication context is processed depending on the comparison attribute sent in the authentication context request.

SP-requested Authentication Context	Comparison Attribute Value	IdP-configured Authentication Context	Status Response
Password	exact	InternetProtocol	NoAuthnContext
Password	minimum	InternetProtocol	NoAuthnContext
Password	better	InternetProtocol	NoAuthnContext
InternetProtocol	exact	InternetProtocol	Success
InternetProtocol	minimum	InternetProtocol	Success
InternetProtocol	maximum	InternetProtocol	Success
InternetProtocol	maximum	Password	NoAuthnContext

SP-requested Authentication Context	Comparison Attribute Value	IdP-configured Authentication Context	Status Response
InternetProtocol	better	Password	Success

Determine Authentication Context and Strength Levels with your Partner

The SP can require specific authentication context classes and strength levels before it permits access to a requested resource. Based on the sensitivity of the resources at the SP, the SP has to feel confident in the assertion it receives from the IdP.

The administrators at the IdP and SP have to establish guidelines for supported authentication contexts and the relative strength of each class. The order of the classes at the IdP together with the associated strength levels affects how it can respond to the SP.

For example, an SP requests an authentication context class of X.509 certificates with a strength level of 3. The IdP has to authenticate the requesting user at a suitable strength level. The comparison value in the request from the SP defines the evaluation of the authentication context. The authentication context that the IdP provides has to satisfy the requirement that is indicated by the comparison. The strength level is an exact match, a minimum or maximum level, or a better strength level.

Configure an Authentication Context Template

An authentication context template defines the specific SAML 2.0 AuthnContext URIs that a partner supports. Each URI identifies the context class.

You can select a template on a per-partnership basis; multiple partnerships can use a single template.

In addition to the common function, a template has distinct functions at each partner:

Authentication Context Template at the IdP

You only require a template at the IdP under the following conditions:

- The SiteMinder Connector is enabled.
- Delegated authentication is the authentication method for single sign-on.
- The IdP automatically detects the authentication context from the SP request.

The template maps URIs to the protection levels associated with a CA SiteMinder® user session. The protection levels indicate the strength of the CA SiteMinder® authentication scheme, from 1 through 1000, with 1000 being the strongest. An administrator assigns protection levels when configuring CA SiteMinder® authentication scheme that authenticates a user and establishes a user session.

Note: Protection levels are only available if you are using the SiteMinder Connector.

Authentication Context Template at the SP

A template at the SP is required to generate an authentication context request. After the SP generates the request, it sends it to the IdP. The template is also required for the SP to validate that the assertion from the IdP satisfies the authentication context request.

Set up an Authentication Context Template

Set up an authentication context template to implement authentication context processing. This procedure is the same for an Identity Provider or Service Provider.

Follow these steps:

1. Log in to the Administrative UI.
2. From the Federation tab, select AuthnContext Templates.

   The View Authentication Context Templates window opens.
3. Select Create Template.

   The template wizard opens at the first step.
4. Enter a name for the template.
5. Complete one of the following actions:
   - Manually enter a URI and click Add URI.
   - Click Load Default URIs to select URIs from a predefined list. Move URIs from the Available URIs to the Selected URIs list.
6. Arrange the selected URIs by strength level. The strength level is in descending order, with the strongest URI at the top and the least strong at the bottom.
7. Click Next.
8. (Optional) Group URIs that require the same level of strength indenting one URI under the previous URI. Use the Change Grouping arrow to move a URI into or out of a group.

9. For SiteMinder Connector deployments only:
   a. Click Enable Protection Levels.
   b. Map the protection levels from a CA SiteMinder® authentication scheme to the URIs. CA SiteMinder® protection levels indicate the strength of an authentication scheme, ranging between 1 through 1000, with 1000 being the strongest. Individual URIs can have unique protection levels; however, grouping URIs means that they have the same level of strength.

      Consider the following information when assigning protection levels:

      - Assign the protection levels in descending order. List the strongest context at the top and the weakest context at the bottom.
      - You can modify the maximum protection level and the Administrative UI calculates the minimum. The Administrative UI verifies that there is no gap in the range of levels so that each protection level has an associated URI.

   Read more about protection level assignments.
10. Click Next to move to the last step of the wizard.
11. Select Finish to confirm the configuration.

The template is complete.

Protection Level Assignments for a Context Template

A federation deployment that uses the SiteMinder Connector for delegated authentication requires that you associate protection levels with each authentication URI. The protection level indicates a level of assurance in the strength of the authentication. Each protection level is mapped to a URI strength level. Ensure that the protection level assignments reflect the protection levels of the CA SiteMinder® authentication scheme.

Note: In a deployment with the SiteMinder Connector, the protection level overrides the level specified in the connector authentication scheme.

When you assign protection levels in the Administrative UI, specify a range. Specify the maximum level for each URI in the list. The minimum protection level is automatically calculated based on the maximum level for the subsequent URI in the list. The range has to cover the configured CA SiteMinder® authentication schemes. For example, if CA SiteMinder® configures an X.509 authentication scheme at a protection level of 20, ensure that the range specified for CA SiteMinder® Federation Standalone includes 20.

Protection Level Example

SiteMinder Authentication Scheme	Protection Level
urn:oasis:names:tc:SAML:2.0:ac:classes:X509	20
urn:oasis:names:tc:SAML:2.0:ac:classes:MobileTwoFactorContract	15
urn:oasis:names:tc:SAML:2.0:ac:classes:InternetProtocol	10
urn:oasis:names:tc:SAML:2.0:ac:classes:Password	5

Each protection level is mapped to a URI strength level. The table shows the original list of URIs:

URI	Protection Level Max	URI Strength
urn:oasis:names:tc:SAML:2.0:ac:classes:X509	1000	4
urn:oasis:names:tc:SAML:2.0:ac:classes:MobileTwoFactorContract	15	3
urn:oasis:names:tc:SAML:2.0:ac:classes:InternetProtocol	10	2
urn:oasis:names:tc:SAML:2.0:ac:classes:Password	5	1

The ranges cover the protection level of the CA SiteMinder® authentication scheme. For example:

- X509 scheme covers protection levels 16-1000
- MobileTwoFactorContract covers protection levels 11-15
- Internet Protocol covers 6-10
- Password covers 1-5

If you group several of the URIs, the grouping enables URIs with different protection levels to have the same URI strength. The following modified table shows the groupings.

URI	Protection Level Max	URI Strength
urn:oasis:names:tc:SAML:2.0:ac:classes:X509	1000	3
urn:oasis:names:tc:SAML:2.0:ac:classes:MobileTwoFactorContract	800	3
urn:oasis:names:tc:SAML:2.0:ac:classes:InternetProtocol	700	2

URI	Protection Level Max	URI Strength
urn:oasis:names:tc:SAML:2.0:ac:classes:Password	200	1

The range of strength levels reflects the total number of groups in the list. For example, if there are three groups, the strength level ranges from 1 to the total number groups, which is 3.

Configure Authentication Context Processing at the IdP

The CA SiteMinder® Federation Standalone IdP can obtain the authentication context for an assertion in two ways:

- Use a predefined authentication class

  Specify a URI for the authentication class and ignore the context request from the SP. A hard-coded entry can act as the default authentication context for IdP-initiated single sign-on.

- Detect the authentication class automatically—only available with the SiteMinder Connector enabled.

  The system automatically detects the authentication context using the authentication context template.

  The IdP uses the template even if the authentication request from the SP does not include the <RequestedAuthnContext> element. The presence of the element triggers extra evaluation by the IdP and constrains the choices of what the IdP puts in the assertion.

  You can find more information about the flow of authentication context processing (see page 262).

Specify How the IdP Obtains the Authentication Context

Configure how to obtain the authentication context.

Follow these steps:

1. Navigate to the SSO and SLO step in the IdP->SP partnership wizard.
2. In the Authentication section, specify how to obtain the authentication context.
   - For local authentication, you are required to use the predefined authentication class.
   - For delegated authentication with the SiteMinder Connector, select the predefined authentication class or automatically detect the class with an authentication context template.

3. Follow the steps for the method chosen in the previous step:
   - To include a predefined class in the assertion, select a URI from the Authentication Class pull-down menu.
   - To include a class from the session context and a template, select a template from the authentication Context Template field or click Create Template.

     Note: This option is available only if you enabled the SiteMinder Connector.
4. (Optional). Depending on how you obtain the authentication context you can also select the Ignore RequestedAuthnContext check box.

The following table shows how the Configure AuthnContext and the Ignore RequestedAuthnContext settings work together:

Configure AuthnContext	Ignore RequestedAuthnContext	SP requests AuthnContext	Result
Predefined Class	Selected	Yes	IdP ignores the <RequestedAuthnContext> and uses the defined value in the assertion.
Predefined Class	Selected	No	IdP returns the defined value in the assertion by default.
Predefined Class	Not selected	Yes	Transaction fails because the IdP is not configured to handle the authentication context request. The IdP returns an error message to the SP.
Predefined Class	Not selected	No	IdP returns the defined class value in the assertion by default.
Automatically Detect Class	Selected	Yes	IdP compares the protection level for the authentication scheme against the authentication context template and returns the matching authentication URI in the assertion. The IdP ignores the values in the SP request.
Automatically Detect Class	Selected	No	IdP compares the protection level for the authentication scheme against the authentication context template and returns the matching authentication URI in the assertion. The IdP ignores the values in the SP request.
Automatically Detect Class	Not selected	Yes	IdP compares the protection level against the authentication context class that the SP sends. The IdP uses the authentication context template to determine the authentication URI it places in the assertion.

Configure AuthnContext	Ignore RequestedAuthnContext	SP requests AuthnContext	Result
Automatically Detect Class	Not selected	No	IdP compares the protection level for the authentication scheme against the authentication context template and returns the matching authentication URI in the assertion.

Configure Authentication Context Requests at the SP

The authentication context is part of an assertion authentication statement and it indicates how a user authenticated at an IdP. An SP can require information about the authentication process to establish a level of confidence in the assertion before granting access to resources.

Authentication Context URIs are the value of the <AuthnContextClassRef> element inside of a <AuthnContext> element. Each URI identifies the context class that the SP wants the IdP to return in the assertion.

The authentication context template at the SP defines the following information:

- Which URIs the SP wants to receive from the IdP. For outgoing requests, the URIs in the template indicate which authentication contexts are acceptable to the SP before it allows access to the requested resource.
- How the URIs in the request are compared to the URIs defined at the IdP.
- How the SP uses the URIs. The SP can include URIs in the outgoing authentication request. The SP can also validate URIs in the incoming assertion response. You can configure the URI usage for both functions.

You can select a template on a per-partnership basis and multiple partnerships can use a single template.

Create an authentication context template before you enable authentication context requests or while you are configuring the SP partnership.

Enable Authentication Context Requests at the SP

An SP can request that an IdP return the authentication context in an assertion. Enable that request at the SP->IdP partnership.

Before you begin, we recommend that you create an authentication context template.

Follow these steps:

1. Log in to the Administrative UI.
2. Select
3. Select the SP->IdP partnership you want to edit.
4. Navigate to the Configure AuthnContext step in the partnership wizard.

   The configuration dialog opens.
5. Select the Enable Authentication Context Processing check box.
6. Complete the fields in the dialog.

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

   Note the following information:

   - If no authentication context template exists, select Create template.
   - The Comparison field describes how the URIs in the SP authentication request are compared with the URIs configured at the Identity Provider.

     The Help details each comparison operator.
   - If you are selecting URIs from the Available URIs list, the available URIs reflect the URIs configured for the chosen template. If there are no predefined templates, click Create Template to configure one.

The authentication context request is included in the authentication requests sent to the Identity Provider.

Chapter 18: Sign and Encrypt Federation Messages

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

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

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

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

This section contains the following topics:

Signature Configuration at a SAML 1.1 Producer and WSFED IP (see page 273)
Signature Verification at a SAML 1.1 Consumer and a WSFED RP (see page 274)
Signature Configuration at a SAML 2.0 IdP (see page 275)
Encryption Configuration at a SAML 2.0 IdP (see page 276)
Signature Configuration at a SAML 2.0 SP (see page 277)
Encryption Configuration at a SAML 2.0 SP (see page 279)

Signature Configuration at a SAML 1.1 Producer and WSFED IP

The Signature step lets you define how the Policy Server uses private keys and certificates to sign SAML assertion or WS-Federation token responses. For SAML 1.1, you can elect to sign only assertions instead of the assertion response.

Note: SAML 1.1 and WS-Federation do not support encryption.

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

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

Follow these steps:

1. Log in to the Administrative UI
2. Select the asserting-to-relying party partnership that you want to modify.
3. Navigate to the Signature step in the partnership wizard.
4. In the Signature section, select an alias from the pull-down list for the Signing Private Key Alias field.

   If there is no private key in the certificate data store, click Import to import a key. Alternatively, click Generate to create a certificate request.

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

   Note: Click Help for a description of fields, controls, and their respective requirements.
5. (SAML 1.1 only) For the Artifact and Post signature options, select the specific components (assertion, response) that you want signed.

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

Signature configuration is complete.

Signature Verification at a SAML 1.1 Consumer and a WSFED RP

The Signature step lets you define how the Policy Server uses private keys and certificates to verify SAML assertion or WS-Federation token responses. For SAML 1.1, you can elect to verify only assertions.

Note: SAML 1.1 and WS-Federation do not support encryption.

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

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

Follow these steps:

1. Log in to the Administrative UI
2. Select the relying-to-asserting party partnership that you want to modify.
3. Navigate to the Signature step in the partnership wizard.

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

   By completing this field, you are indicating which certificate verifies signed assertions or responses or both. If there is no certificate in the certificate data store, click Import to import one. Alternatively, click Generate to create a certificate request.

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

Signature configuration is complete.

Signature Configuration at a SAML 2.0 IdP

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

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

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

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

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

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

To configure signing options

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

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

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

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

   Select the algorithm that best suits your application.

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

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

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

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

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

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

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

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

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

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

Encryption Configuration at a SAML 2.0 IdP

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

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

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

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

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

To configure encryption options

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

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

   The W3C XML Syntax and Processing standards define these algorithms.

   Important! For the following block/key algorithm combinations, the minimum key size that is required for the certificate is 1024 bits.

   - Encryption Block Algorithm: 3DES

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

     Encryption Key Algorithm: RSA-OEAP

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

The encryption configuration is complete.

Signature Configuration at a SAML 2.0 SP

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

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

  Note: For SAML 2.0 POST binding, the IdP is required to sign assertions.
- Sign single logout responses and requests (HTTP-Redirect and SOAP bindings).

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

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

To configure signing options

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

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

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

   Select the algorithm that best suits your application.

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

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

   By completing this field, you are indicating which certificate the relying party uses to verify signed assertions or single logout requests and responses. If there is no certificate in the database, click Import to import one.
5. (Optional) For the SP to sign all authentication requests, select the Sign Authentication Requests. If the remote asserting party requires the authentication requests to be signed, check this option.

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

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

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

Encryption Configuration at a SAML 2.0 SP

The Signature and Encryption step lets you configure how the SP uses private keys and certificates, including encrypting and decrypting assertions, Name IDs, and attributes.

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

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

To configure encryption options

1. In the Encryption section, select one or both of the following check boxes so that the correct data is encrypted in the assertion:
   - Require encrypted Name ID
   - Require encrypted Assertion

   Note: To use the AES-256 bit encryption block algorithm, install the Sun Java Cryptography Extension (JCE) Unlimited Strength Jurisdiction Policy Files. You can download these files from http://java.sun.com/javase/downloads/index.jsp.
2. Select the alias from the certificate data store for the Decryption Private Key Alias.

   This private key decrypts any encrypted assertion data. If no certificate available, click Import to import one or click Generate to create a key pair and generate a certificate request.

The encryption configuration is complete.

Chapter 19: Session Duration Management at a Service Provider

This section contains the following topics:

How to Manage the Authentication Session Duration at a Service Provider (see page 281)

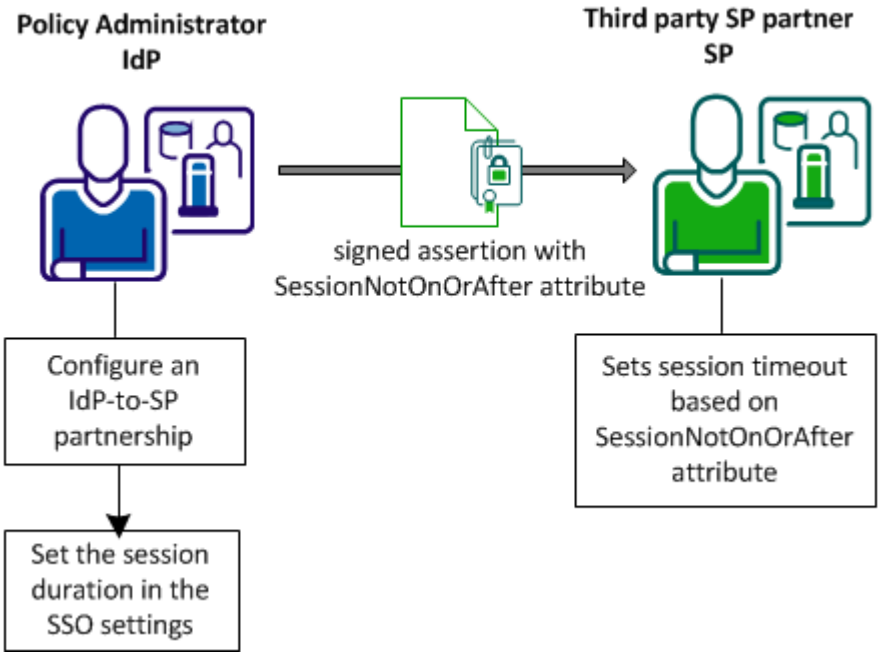
How to Manage the Authentication Session Duration at a Service Provider

You can manage the duration of the authentication session at the Service Provider. The SessionNotOnOrAfter attribute is an optional attribute that the IdP can include in the <AuthnStatement> of an assertion.

Note: The SessionNotOnOrAfter parameter is different from the NotOnOrAfter parameter, which determines how long the assertion is valid.

The value of determining session duration is to prevent a user from authenticating again if the session at the SP is too brief. A third-party SP can use the value of the SessionNotOnOrAfter to set its own timeout values, helping to ensure that sessions are not too short. If a user session becomes invalid, the user has to reauthenticate at the Identity Provider. To create a seemless experience for the user, manage the sessions at the SP accordingly.

The following graphic shows the configuration steps at the IdP and the resulting action that the third-party SP takes.



Include a Session Duration Attribute in an Assertion

The configuration for session duration is done at the IdP. The assertion sent to the SP includes the session attribute that the SP uses to set timeout values for SP site.

Important! If CA SiteMinder® Federation Standalone is acting as an SP, it ignores the SessionNotOnOrAfter value. Instead, the SP sets session timeouts from the realm timeout that corresponds to the SAML authentication scheme protecting the target resource.

Follow these steps:

1. Log in to the Administrative UI.
2. Select the IdP->SP partnership you want to modify.
3. Navigate to the SSO and SLO step.

4. In the SSO section, select the option for the Recommended SP Session Duration. If you select the customize option, you can select one of the following options:
   - omit the attribute
   - set the attribute to the IdP session timeout
   - specify your own duration period

   Note: Click Help for a description of fields, controls, and their respective requirements.
5. Select the Confirm step after you complete your changes and click Finish.

Based on the configuration, a session attribute is placed in the assertion and sent to the SP.

Chapter 20: CA SiteMinder® Integration with CA SiteMinder® Federation Standalone

This section contains the following topics:

How to Integrate CA SiteMinder® Federation Standalone and CA SiteMinder® (see page 285)

How to Integrate CA SiteMinder® Federation Standalone and CA SiteMinder®

A deployed CA SiteMinder® system can integrate with CA SiteMinder® Federation Standalone using the CA SiteMinder® Connector, a software component included with CA SiteMinder® Federation Standalone. The Connector enables the following interaction between federation and web access management deployments:

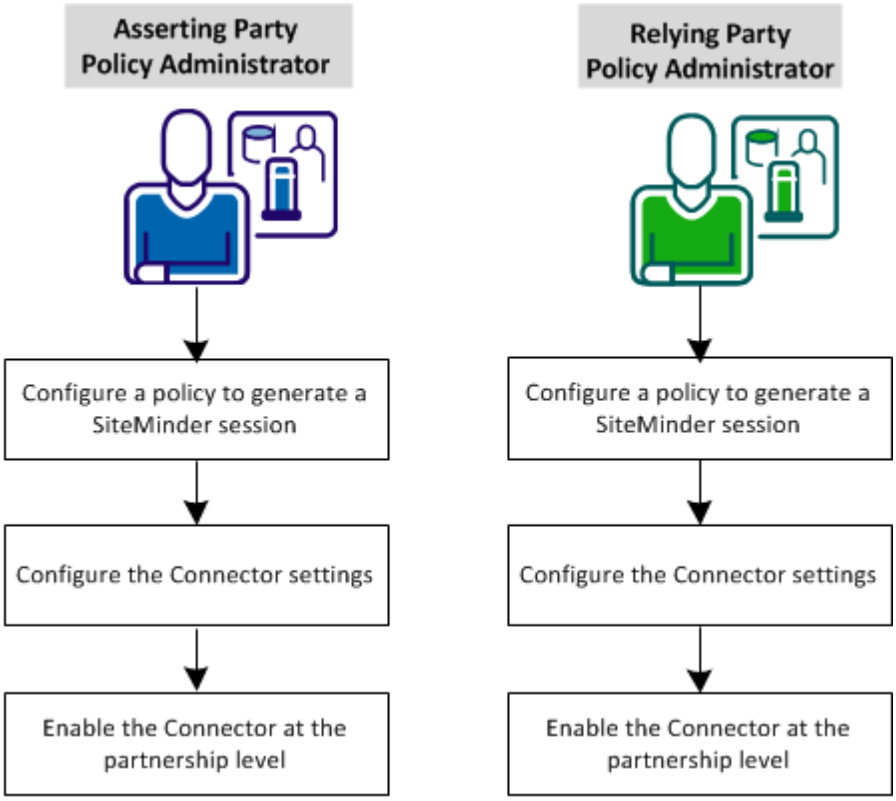
- Establishing an identity to generate an assertion.

  At the asserting party, a user arrives at CA SiteMinder® Federation Standalone but has no session. The Connector communicates with CA SiteMinder® to establish a CA SiteMinder® session. Based on the contents of the CA SiteMinder® session, the Connector creates a federation session. Using this session information, a SAML assertion is generated for the user.

- Establishing an identity for CA SiteMinder® to determine authorization privileges.

  At the relying party, a user authenticates with CA SiteMinder® Federation Standalone and a federation session is generated. The Connector passes on the federation session with the user name to CA SiteMinder®, which generates a CA SiteMinder® session from the federation session. The user is now identified and is not rechallenged for credentials. CA SiteMinder® determines the authorization privileges for the requested resources at the relying party.

The following figure shows the configuration process when integrating with the Connector:



Complete the following configuration steps:

1. Configure a policy to generate a CA SiteMinder® session.
2. Configure the Connector settings.
3. Enable the Connector at the partnership level.

Integrate with CA SiteMinder® using the SiteMinder Connector

The CA SiteMinder® Connector enables the following integrations:

- Establishing an identity to generate an assertion.

  At the asserting party, a user arrives at CA SiteMinder® Federation Standalone but has no session. The Connector communicates with CA SiteMinder® to establish a CA SiteMinder® session. The use of the session information results in a federation session and the generation of a SAML assertion for the user. With this assertion, the user can access protected federated resources at the relying party.

- Determining authorization privileges from the assertion.

  At the relying party, a user authenticates with CA SiteMinder® Federation Standalone and a federation session is generated. The Connector passes on the federation session with the user name to CA SiteMinder®, which generates a CA SiteMinder® session. By establishing this session, these users do not get rechallenged when accessing a protected resource. The user is now identified and access privileges for the user at the relying party can be determined.

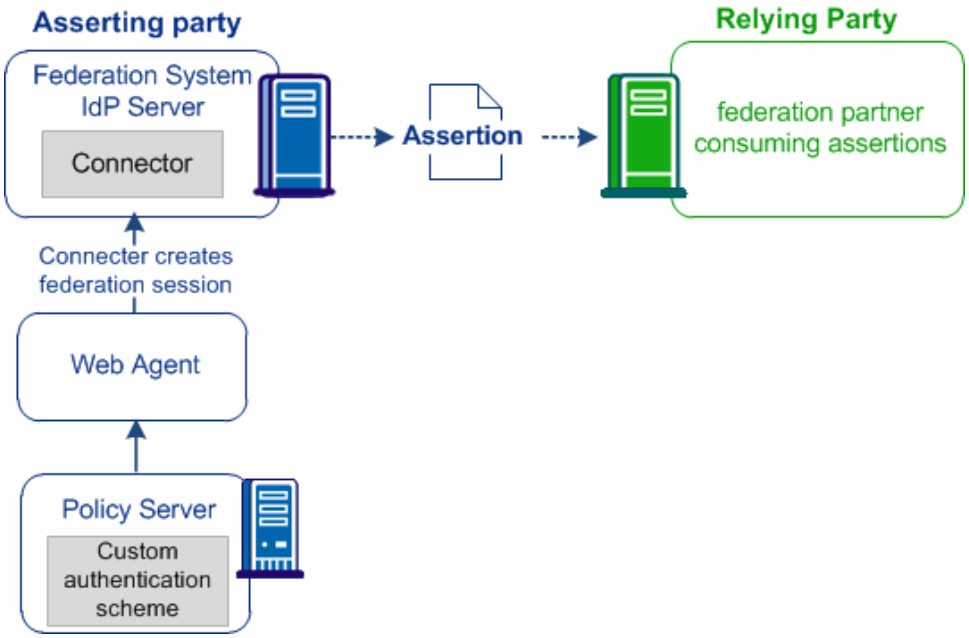
The FEDSESSION cookie uses the following timeout settings:

- Idle Timeout: 600 seconds (10 minutes)
- Max Timeout: 900 seconds (15 minutes)

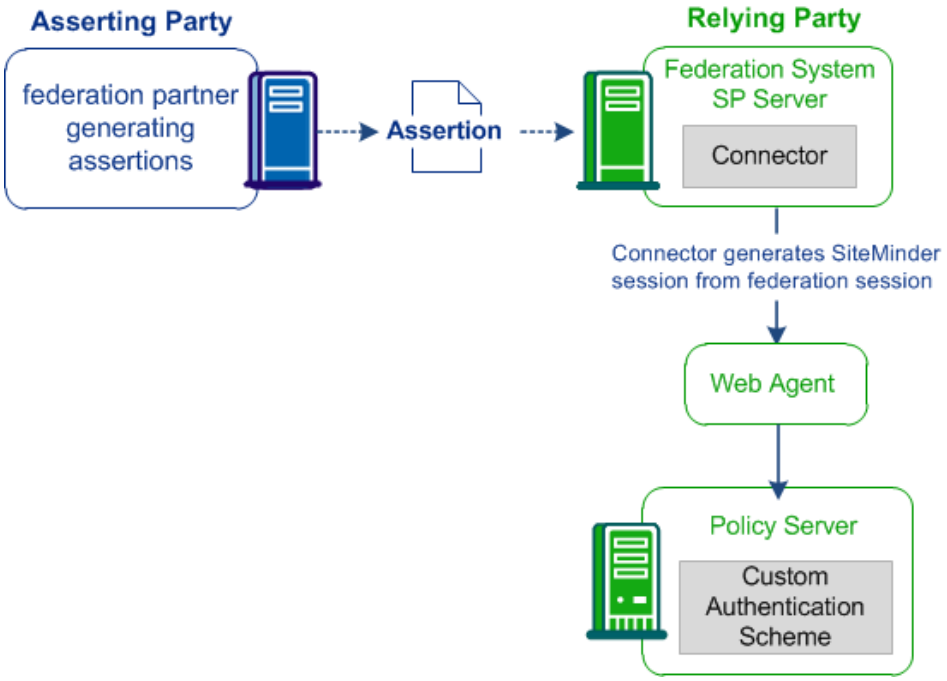
You cannot change these timeout settings in UI.

The Connector requires configuration in the CA SiteMinder® environment and in the CA SiteMinder® Federation Standalone environment, as shown in the following diagrams.

This graphic shows the Connector at the asserting party:



This figure shows the Connector at the relying party.



Configure a Policy to Generate a Session at Each Site

The CA SiteMinder® Connector enables CA SiteMinder® Federation Standalone to work with an existing Policy Server. The first step is to configure a policy. At the asserting party, the policy generates a federation session. At the relying party, the policy generates a CA SiteMinder® session. Though this policy functions as any other policy, its main objective is to trigger a session, not to protect resources.

Note: Configure a policy at the asserting and the relying party.

The policy requires that you configure the typical policy objects; however, you apply a custom CA SiteMinder® Connector authentication scheme. This policy is specific to the Connector setup.

To configure the Policy Server objects, see the *Policy Server Configuration Guide.*

Important! Complete the following steps at the Policy Server before configuring the Connector.

Follow these steps:

1. Unzip the smauthconnectors.zip archive on your federation system. This archive is included with the federation product kit.
2. Select the correct custom authentication scheme library for your CA SiteMinder® operating environment:
   - **Windows:** smauthsmconnector.dll
   - **Solaris/Linux:** libsmauthsmconnector.so

     Note: The name is case-sensitive on UNIX platforms.
3. Copy the library to the appropriate Policy Server directory on the CA SiteMinder® system:
   - **Windows:** *policy_server_home*/siteminder/bin
   - **Solaris/Linux:** *policy_server_home*/siteminder/lib
4. Log on to the CA SiteMinder® Administrative UI.
5. Create a Web Agent that represents the federation system. For example, name it Federation Agent.

   Important! Do not select the option for supporting 4.x agents.
6. Create an Agent Configuration Object, which specifies the Agent configuration, and specify a value for the DefaultAgentName setting. This setting alone is sufficient for the object.
7. Create a Host Configuration Object.

   The Host Configuration Object defines the connection between a trusted host and the Policy Server. To integrate the federation system and the Policy Server, the Host Configuration Object defines the Policy Server to which the federation system can connect.

   For the federation system to connect to one or more Policy Servers in an existing Host Configuration Object, use that object. Otherwise, create one for the federation-to-Policy Server-connection.

8. Create a custom Connector authentication scheme with the following values:

   Library

   smauthsmconnector

   This value is case-sensitive.

   Secret

   alphanumeric string

   The value for this field must match the value Shared Secret value in the Connector settings in the Administrative UI.

9. Create a policy domain for the federation product. This domain must contain the necessary realm and resource that you add to the policy to create a CA SiteMinder® session.

10. Add the user directory that is used by the federation system and the Policy Server to the domain you configured.

11. Create a realm with the following values:

    Agent

    Specify the Web Agent from the previous step.

    Resource Filter

    Specify a dummy directory, such as /federation/. This directory does not have to exist on a web server.

    Authentication Scheme

    Enter the name that you gave to the custom authentication scheme created previously.

12. Create a rule with the following values:

    Resource

    *

    Action

    Web Agent—Get and Post

13. Create a policy with the following settings:

    Users

    Specify the users from the user directory that the federation system and CA SiteMinder® share.

    Rules

    Add the rule that is created for the Connector.

You now have a policy that generates a CA SiteMinder® session when communicating with CA SiteMinder® Federation Standalone.

Configure the Connector Settings

For the Connector to interact with CA SiteMinder®, configure the Connector settings in the CA SiteMinder® Federation Standalone Administrative UI. All partnerships that use the Connector use a single configuration and connect to a single CA SiteMinder® environment.

Follow these steps:

1. Log in to the Administrative UI.
2. Navigate to the Infrastructure tab.
3. Select Deployment Settings.

   The Configure Deployment Settings dialog opens.
4. Fill in all the fields in the CA SiteMinder® Connector Settings section. Note the following considerations:

   - To enable or disable the Connector for a given partnership, enable it at the partnership level.
   - To enable or disable the Connector globally, use the check box in the deployment settings.

   Important! If the Connector is disabled at the global level, CA SiteMinder® Federation Standalone ignores the check box at the partnership level.

   Note: Click Help for a description of fields, controls, and their respective requirements.
5. Select Register Host and provide the legacy administrator credentials for the CA SiteMinder® Policy Server. Only legacy administrators can perform host registration.

   This step registers CA SiteMinder® Federation Standalone as an Agent with the CA SiteMinder® Policy Server.

   Note: You can configure failover support for the host registration process by specifying more than one Policy Server. If the registration with the primary Policy Server fails, the registration process tries with the next Policy Server specified until the registration process completes successfully.
6. Click Save.

   Important! Select Save specifically in the CA SiteMinder® Connector Settings section after registering the host.

7. Restart the federation services according to your operating environment.

   - **Windows**

     Use the stop and start shortcuts as follows. If you logged in as a network user and not a local administrator, right-click the shortcut and select Run as administrator.

     a. Start, All Programs, CA, Federation Standalone, Stop services

     b. Start, All Programs, CA, Federation Standalone, Start services

   - **UNIX**

     a. Open a command window.

     b. Run the following scripts:

     federation_install_dir/fedmanager.sh stop

     federation_install_dir/fedmanager.sh start

     Note: Do not stop and start the services as the root user.

The CA SiteMinder® Connector configuration is complete.

Enable the Connector at the Partnership Level

Before you enable the Connector, verify:

- The CA SiteMinder® Policy Administrator has configured the policy for federated communication.
- You have configured the Connector-specific settings in CA SiteMinder® Federation Standalone.

Enable the Connector for the partnership where CA SiteMinder® is deployed:

- If CA SiteMinder® is at the asserting party, enable the Connector for an IdP-to-SP or Producer-to-Consumer partnership.
- If CA SiteMinder® is at the relying party, enable the Connector for an SP-to-IdP or Consumer-to-Producer partnership.

Whether you are modifying an existing partnership of configuring a new partnership, the standard partnership configuration steps apply; there are no unique configuration procedures. However, specify the target resources at the relying party in the using the following guidelines:

- If CA SiteMinder® Federation Standalone is deployed in standalone mode, the target resource resides on the web server that the CA SiteMinder® Web Agent protects.
- If CA SiteMinder® Federation Standalone is deployed in proxy mode, the target resource is the URL for the CA SiteMinder® Federation Standalone server because all proxy requests go back to CA SiteMinder®.

Follow these steps:

1. Log in to the Administrative UI.
2. Select a partnership from the Federated Partnerships list or create a new one.

   The Partnership dialog opens.
3. Navigate to one of the following steps in the wizard:
   a. At the relying party, navigate to the User Identification step in the Partnership wizard.
   b. At the asserting party, navigate to the Federation Users step in the Partnership wizard.
4. Select the Enable SiteMinder Connector check box.

   The configuration fields become available.
5. (Optional) Select the Enforce UserDN and Directory Name Comparison check box. Selecting this check box forces a comparison of the UserDN and UserDirectory Name entries between the user directory at CA SiteMinder® Federation Standalone and the directory at CA SiteMinder®.

   If you select this check box, the user directory for the CA SiteMinder® Federation Standalone and the CA SiteMinder® deployment must be the same physical directory. The name for both of these directories must be the same for user store lookups. If you clear the check box, the Universal ID is the attribute that finds the user record. If the Universal ID is used, the directories do not have to be the same. If you rely on the Universal ID, each user must have a unique Universal ID. If the Universal IDs are not unique, the system accessing the user record can retrieve the wrong record.
6. Save your changes.

To disable the Connector, you can do so at the partnership level or globally in the Deployment Settings.

Chapter 21: Secure a Federated Environment

This section contains the following topics:

Protecting Federated Communication (see page 295)

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® Federation Standalone, here are some recommendations for protecting your environment:

- Generating assertions for only one time use.
- Securing connections across the federated environment.
- Protecting against cross-site scripting.

These topics are described in the sections that follow.

Enforcing the One Time Use of an Assertion

Reusing an assertion beyond its validity results in authentication decisions based on out-of-date identity information. To prevent reuse, CA SiteMinder® Federation Standalone can generate an assertion intended for one-time use, in compliance with the SAML 1.x and 2.0 specifications. The assertion contains elements that tell the relying party not to retain the assertion for future transactions, preventing problems associated with reusing an assertion.

If CA SiteMinder® Federation Standalone is acting as the asserting party (Producer/IdP), you can configure the one time use of an assertion. For a SAML 1.x producer, 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® Federation Standalone 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. CA SiteMinder® Federation Standalone uses the single use policy when acting as the relying party, and it is only for POST transactions. The one time use feature is for HTTP-Artifact and HTTP-POST.

Securing Connections Across the Federated Environment

Identity information 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

It is important to secure data transmission from the relying party to the target application at the client site. 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.

Protect against this vulnerability by setting the Enable Secure Cookies check box in the Deployment Settings of the Administrative UI. The Enable Secure Cookies setting instructs CA SiteMinder® Federation Standalone to generate cookies marked with the "secure" flag. This flag indicates that CA SiteMinder® Federation Standalone sends the cookie only over an SSL communication channel.

Securing the Initial Authentication at the CA SiteMinder® Federation Standalone Asserting Party

The initial authentication of a user at a CA SiteMinder® Federation Standalone 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 steal sensitive user information for impersonation or identity theft.

Protect against this vulnerability by setting the Enable Secure Cookies check box in the Deployment Settings of the Administrative UI. The Enable Secure Cookies setting instructs CA SiteMinder® Federation Standalone to generate cookies 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 a Federated Network Against Cross-Site Scripting

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

CA SiteMinder® Federation Standalone provides several JSPs for use with 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® Federation Standalone receives a 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. This is a sample application you can use to direct the user first 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
)	closed (right) parenthesis
&	ampersand
+	plus sign

Each CA SiteMinder® Federation Standalone-provided JSP contains a variable that defines the characters to scan. Modify these JSPs to expand the character set.

Chapter 22: Application Integration at the Relying Party

This section contains the following topics:

Relying Party Interaction with Applications (see page 299)
Redirect a User to the Target Application (see page 299)
Using HTTP Headers to Pass Assertion Data (SAML only) (see page 301)
Mapping Assertion Attributes to Application Attributes (SAML Only) (see page 302)
Dynamic Provisioning of a User Identity at the Relying Party (see page 308)
Failed Authentication Handling Using Redirect URLs (Relying Party) (see page 316)

Relying Party Interaction with Applications

The Application Integration step of the partnership wizard is applicable only at the relying party. This step lets you define various aspects of federated operation for resolving user identities and directing users to the target application.

The features that you can configure in the Application Integration step are:

- User redirection to the target application
- Mapping assertion attributes to application attributes (SAML only)
- Provisioning a user identity
- User redirection in case of an authentication failure

Redirect a User to the Target Application

The Target Application group box in the Application Integration step lets you define how a user gets redirected from CA SiteMinder® Federation Standalone to the target application. There are several methods from which to choose. The redirection method you select depends on the type of data you want to pass along with the user to the target application.

Follow these steps:

1. Navigate to the Application Integration step in the Partnership wizard.
2. Select a redirection method for the Redirect Mode field.
   - If you select Cookie Data, you can URL-encode attribute data in the cookie by selecting the URL Encode Attribute Cookie Data check box.
   - If you select open format cookie as the mode, there are additional settings you must configure, such as the name of the cookie, the algorithm that encrypts the cookie along with the encryption password. Optionally, you can enable an HMAC function to verify the integrity of the cookie.

     If the relying party receives an assertion with multiple attribute values, CA SiteMinder® Federation Standalone passes all values to the target application in the cookie.
   - If you select one of the FIPS-compatible algorithms (AES algorithms), you are required to use a CA SiteMinder® Federation Standalone SDK to consume the open format cookie. If you use the .NET SDK, you are required to use the AES128/CBC/PKCS5Padding encryption algorithm.
   - If you configured CA SiteMinder® Federation Standalone in proxy mode and you select HTTP Headers as the redirect mode, CA SiteMinder® Federation Standalone can deliver multiple attribute values in a single header by separating each value with a comma.

   Note: Click Help for a description of fields, controls, and their respective requirements.
3. Enter the URL of the target application in the Target field.

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

   Note: You specify the backend server that sits behind the proxy host when you run the CA SiteMinder® Federation Standalone Configuration wizard. You can modify the backend server entry by rerunning the Configuration wizard.

   For SAML 2.0, you can leave this field blank if you override it with the value of the Relay State query parameter, which can be included in a URL to trigger single sign-on. To enable this override, select the Relay state overrides target check box.

Setting up redirection to the target is complete.

Using HTTP Headers to Pass Assertion Data (SAML only)

For a SAML entity, the Policy Server can use HTTP headers to pass identity attributes from an assertion to a back-end application. A backend application can be a target application for single sign-on or a user provisioning application. The system passes these headers in an encrypted cookie.

The headers have the same name as the assertion attributes. For example, if the assertion attribute is "address", the application looks for the HTTP header "ADDRESS".

Assertion attributes are case-sensitive, but HTTP headers are not. The Policy Server cannot pass the same attributes that differ only by case sensitivity and then map them to HTTP headers. For example, the system cannot pass "address" and "Address" as headers at the same time. In general, do not use the attributes with the same names that are only different because of case sensitivity or format.

The following additional values are passed as headers:

- NAMEID
- FORMAT
- AUTHNCONTEXT

Protecting HTTP Headers

If an unauthorized user knows the name of an assertion attribute, that user can set this name as a header in a browser. With the header set, the malicious user can gain access to the target application. The target application sees an expected header value and grants access to the resource without CA SiteMinder® consuming an assertion.

Setting a value for the FedHeaderPrefix protects against the following scenario:

1. An unauthorized user learns the names of HTTP headers. These header names include prefixes.
2. The malicious user sends an incoming request, including the headers, to the Policy Server.
3. The Policy Server recognizes that the headers containing prefixes come from an incoming request and are not generated internally so it removes these headers.
4. Before the system passes its own legitimate headers to the back-end application, it adds the specified prefix to each header. The headers are then passed to the application.

Configure HTTP Headers to Pass Assertion Data (SAML only)

CA SiteMinder® can pass assertion data using the HTTP headers.

Follow these steps:

1. Verify that the CA SiteMinder® web agent is installed on the relying party system that is handling federation traffic.
2. (Optional but recommended) Enter any string as a prefix for the HTTP header.

   CA SiteMinder® adds the prefix to all the HTTP headers. Setting a prefix protects HTTP headers against manipulation by an unauthorized user before CA SiteMinder® consumes an assertion. As a result, only legitimate headers get passed to the target application. Read more about protecting HTTP headers (see page 301).

   To add a prefix to the HTTP header, do the following:

   a. Log in to the Administrative UI.

   b. Click Infrastructure, Deployment Settings.

   c. Specify a string for the HTTP Header Prefix.

      Note: This option is available only for the proxy deployment mode.

   d. Click Save.
3. Do *one* of the following tasks in the Application Integration step of the partnership wizard:

   - Select HTTP Headers as the Redirect Mode for the target application.
   - Select HTTP Headers as the Delivery Option for user provisioning.

HTTP headers are now configured to pass attribute data.

Mapping Assertion Attributes to Application Attributes (SAML Only)

At the relying party, CA SiteMinder® Federation Standalone enables you to map a set of assertion attributes to a set of outgoing application attributes. CA SiteMinder® Federation Standalone then delivers the application attributes to the target application. Attribute mapping allows you to provide a customized experience for users without having to modify the target application. Attributes are mapped on a per-partnership basis, which allows you to use a relying party-side application for multiple asserting parties.

CA SiteMinder® Federation Standalone can perform the following types of mapping:

- Convert assertion attribute names to application attribute names.

  Example

  An incoming assertion attribute can be Region=US. The attribute can be converted to an outgoing application attribute ServiceLocation=US.

- Transform separate attributes and their values into a single attribute.

  Example

  Two attributes are included in the assertion, Name=Bob and LastName=Smith. These two attributes can be converted to FullName =Bob Smith.

Using the Application Attributes Definitions Table

You define attribute mapping rules in the Application Attributes Definitions table of the Application Integration dialog.

The Application Attribute and Assertion Attribute columns are populated based on the assertion attributes that are specified for the remote Producer or IdP entity. You configure these attributes at this local relying party. The assertion attribute name is entered for the Application Attribute column. The equivalent Unified Expression Language (UEL) string is entered in the Assertion Attribute(s) column.

Administrators or application integrators at the relying party must know the following information to configure attribute mapping:

- Names of the target application attributes.
- Names of the attributes in the assertion.
- Mapping relationship between the assertion attributes and the target application attributes. Understanding the mapping relationship means that you know how to transform the available assertion attributes into the required application attributes.

Gather the names of the application and assertion attributes from the necessary parties before setting up attribute mapping.

The application attributes must reflect the attributes that are used by the target application so you must modify the default values to suit the application. You obtain the application attributes from an out-of-band communication with the application administrator.

Use the Expression Builder to Build Mapping Rules

The UI provides an expression builder to aid in the construction of mapping rules. Access the expression builder by selecting the slider button (<<) to the right of the Assertion Attribute(s) field. The slider button reveals a blank field and pull-down arrow. Select the arrow to see a list of assertion attributes and special characters that you can use to compose a mapping. Click the slider button (>>) to hide the expression builder.

The Assertion Attributes list from the expression builder is pre-populated based on the assertion attributes that are specified for the remote Producer or IdP entity, which you configure at this local relying party. You can specify entries manually as long as you know the attribute is in the assertion. You do not have to use only the options from the expression builder menu.

The Special Characters list contains characters, such as commas and percent signs that you can use to build a mapping rule. You can select a character from the list or can enter the character manually.

Important! When you enter assertion attributes in this table, they are case-sensitive relative to how the assertion attribute is specified at the remote asserting party. The cases must match. If CA SiteMinder® Federation Standalone is at both sides of the partnership, the attributes are specified in the NameID and Attributes step of the remote IdP Partnership wizard. Obtain the assertion attributes in an out-of-band communication with the partner or by importing metadata.

After the mapping rules are defined, CA SiteMinder® Federation Standalone places the data in a legacy cookie, an open format cookie, or an HTTP header and sends the data to the application. You specify the delivery method in the Target Application section of the Application Integration dialog.

Modify and Delete Mappings

You can change or remove attribute mappings in the Application Attributes Definitions table at any time.

To modify a mapping

1. Place your cursor in any of the fields in the row you want to modify and enter the new text. You can also use the expression builder to append additional values to the end of the current expression.
2. Save the change by clicking Next to advance to the end of the wizard.

To delete a mapping

1. Click the trash barrel in the Delete column for the entry you want to remove.
2. Save the change by clicking Next to advance to the end of the wizard.

Construct Attribute Mapping Rules Using the Proper Syntax

Attribute mapping uses mapping rules that transform assertion attributes to application attributes. When attribute mapping is enabled, CA SiteMinder® Federation Standalone generates default mapping rules. The rules are based on the assertion attributes specified for the remote Producer or IdP entity. All this configuration takes place at the local relying party. When attribute mapping is disabled, assertion attributes are passed "as is" to the target application.

CA SiteMinder® Federation Standalone uses a Unified Expression Language (UEL) syntax for mapping that is similar to JSP and JSF. Each assertion attribute is put into a hashmap and assigned the **attr** keyword. A UEL expression evaluator goes through the list of mapping rules and applies them to the hashmap of assertion attributes. The expression evaluator then generates another hashmap containing the resulting application attributes. The hashmap of outgoing application attributes is converted into cookie contents or header variables and delivered to the target application.

For more information about UEL, go to the Sun Developer Network http://developers.sun.com/.

To construct expressions, it is important to understand the syntax CA SiteMinder® Federation Standalone uses for the expressions.

Single Attribute Representation

To represent a single assertion attribute, use the following syntax:

#{attr["*attribute_name*"]}

Example: #{attr["Name"]} represents the value of the Name assertion attribute.

Composite Attribute Representation

Value expressions can be concatenated to form a composite value (with optional delimiter). To represent a composite assertion attribute, use the following syntax:

#{attr["*first_attribute*"]}*optional_character* #{attr["*second_attribute*"]}

Mapping Examples

The following are a series of examples of mapping rules. These examples are presented in the following format:

application_attribute=assertion_attributes_expression

Name Example

Syntax

ID = #{attr["Name"]}

Sample Result

BobSmith

Simple Concatenation Examples

Syntax

FullName = #{attr["FirstName"]},#{attr["LastName"]}

Sample Result

Bob,Smith

Syntax

FullName = #{attr["LastName"]},#{attr["FirstName"]}

Sample Result

Smith,Bob

Spaces are considered special characters. If you want a space between attributes in an expression, enter a space. For example:

Syntax

FullName = #{attr["LastName"]}, #{attr["FirstName"]}

Sample Result

Smith, Bob

Date Examples

Syntax

Date = #{attr["month"]}/#{attr["dateOfMonth"]}/#{attr["year"]}

Sample Result

01/05/2010

Syntax

Date = #{attr["monthSymbol"]} #{attr["dateOfMonth"]}, #{attr["year"]}

Sample Result

January 5, 2010

Monetary Example

Syntax

Price = #{attr["amount"]}#{attr["currency"]}

Sample Result

2.50EUR

Email Address Examples

Syntax

EmailAddress = #{attr["userName"]}@#{attr["domainName"]}

Sample Result

JaneDoe@company.com

Syntax

AcmeEmailAddress = #{attr["AcmeIDKey"]}@acme.com

Sample Result

bsmith@acme.com

Configure Attribute Mapping at the Relying Party

Define a set of mapping rules that CA SiteMinder® Federation Standalone can apply to the assertion attributes. CA SiteMinder® Federation Standalone lets you map a specific assertion attribute or a combination of several attributes. The result of the mapping can be a single application attribute or multiple attributes.

To configure attribute mapping

1. Navigate to the Application Integration step in the Partnership wizard.
2. Select the Enable Attribute Mapping check box in the Map to Application Attribute section.

   An Application Attribute Definitions table displays.
3. Modify any existing application attribute or define new ones in the table. All application attributes are delivered to the target application.

   The syntax of the value in the Assertion Attribute column must comply with Unified Expression Language (UEL).

   Select the slider button (<<) to open the expression builder and display the options available to you. To add the item from the list to the attribute value, select the assertion or special character and click Append.

   Note: When you specify Cookie Data and any special character in the Application Attributes Table, select the URL Encode Attribute Cookie Data option. The check box is in the Target Application section of the dialog. Special characters can be added from the drop-down list or entered manually. Additionally, the target application must URL decodes the name and value of the application attribute received.
4. (Optional) If the default mappings are not sufficient, add as many rows as you like.

   By default, all assertion attributes defined at the remote Producer or IdP entity are included in the table with default (straight) mappings. The original assertion attribute is not changed. You can modify these mappings.
5. Configure the method by which the application attributes are sent to the target application. You configure the method in the Target Application section of the Application Integration dialog.

Attribute mapping configuration is complete.

More information:

Construct Attribute Mapping Rules Using the Proper Syntax (see page 305)

Dynamic Provisioning of a User Identity at the Relying Party

In a federated network, it is not uncommon for the relying party to establish accounts for users federating from different asserting parties. Dynamic provisioning lets CA SiteMinder® Federation Standalone support the process of creating client accounts with the necessary account rights and access privileges for accessing data and applications.

CA SiteMinder® Federation Standalone offers two methods to support provisioning:

- Local account linking (SAML 2.0 only)
- Remote provisioning

Each method is described in the following sections.

Local Account Linking for Provisioning

You can use local account linking to implement provisioning only for SAML 2.0 deployments. Provisioning occurs by linking a user account at the IdP to an account at the SP.

The local account linking process is as follows:

1. A user requests access to a federated target resource at the SP.
2. The SP generates an AuthnRequest that includes an attribute named AllowCreate, which is set to true. The SP sends the AuthnRequest to the IdP to obtain an identity for the user.
3. Upon receiving the AuthnRequest, the IdP generates an assertion. During assertion generation, the IdP searches for the appropriate user record for the attribute that is configured to serve as the Name ID in the assertion.
4. If the IdP cannot find a value for the attribute being used for the NameID, the IdP generates a persistent identifier, assuming the feature to create an identifier is enabled.

   The persistent identifier is a randomly generated ID. The IdP uses this identifier as the value of the Name ID attribute and places it in the assertion. The IdP then returns the assertion to the SP.

   Note: The Allow/Create feature must be configured at both sites. If the IdP is not configured to create an identifier, regardless of whether the Allow/Create attribute is in the AuthnRequest message, assertion generation fails.
5. CA SiteMinder® Federation Standalone at the SP processes the assertion from the IdP, but it cannot find the user record because the NameID value does not exist at the SP. As a result, authentication fails.
6. The failed authentication attempt triggers a redirect to a linkaccount.jsp page with all the assertion and other data that the Assertion Consumer Service passes to it.
7. The user is required to authenticate with local credentials before gaining access to the linkaccount.jsp page. The successful login identifies the user. CA SiteMinder® Federation Standalone references the appropriate user record in the local user directory and creates a session for that user. The user still does not get access to the originally requested federated resource.

8. The linkaccount.jsp passes the assertion and all other data back to the Assertion Consumer Service. CA SiteMinder® Federation Standalone authenticates the user with the assertion again. Now that there is a session that identifies the user, CA SiteMinder® Federation Standalone populates the proper user record in the local user directory with the persistent identifier from the assertion. CA SiteMinder® Federation Standalone stores the persistent identifier in an attribute that you configure specifically for this purpose. The account at the IdP is now linked with the account at the SP. Authentication succeeds.
9. Finally, CA SiteMinder® Federation Standalone redirects the user the requested resource.

Note: Local account linking requires the use of the **Allow IdP to create User Identifier** feature; however, the AllowCreate feature is not exclusively for local account linking. You can select this feature without wanting to implement local account linking.

Configure Local Account Linking (SAML 2.0)

Implementing the local account linking method of provisioning requires configuration at the Identity Provider and Service Provider.

To configure local account linking at the Identity Provider

1. Access the partnership wizard and navigate to the Assertion Configuration step in the partnership wizard.
2. Configure the required fields in the Name ID group box.

   In these fields is where you determine the attribute used for the NameID in the assertion.

   Note: Click Help for a description of fields, controls, and their respective requirements.
3. Select the **Allow Creation of User Identifier** check box.
4. Select the Confirm step the Partnership wizard and click Finish to save your changes.

Configuration at the Identity Provider is complete.

To configure local account linking at the Service Provider

1. Access the Partnership wizard and navigate to the User Identification step.
2. In the Choose Identity Attribute from Assertion group box:
   - Select NameID as the attribute from the assertion used for identification.
   - Select Allow IDP to create user identifier.

3. Enter a value for the Search Specification field.

   The Search Specification value is the attribute CA SiteMinder® Federation Standalone uses to look up the user and to store the persistent identifier sent from the IdP. For example, if buyerID should store the value of the NameID, set the string to buyerID=%s.

4. Navigate to the Application Integration step.

5. Select Local Account Linking for the Provisioning Type field in the User Provisioning section of the dialog.

   Selecting this option automatically configures the User Not Found URL to the linkaccount.jsp page with a method of POST. This URL is where CA SiteMinder® Federation Standalone redirects the user after the first failed authentication attempt.

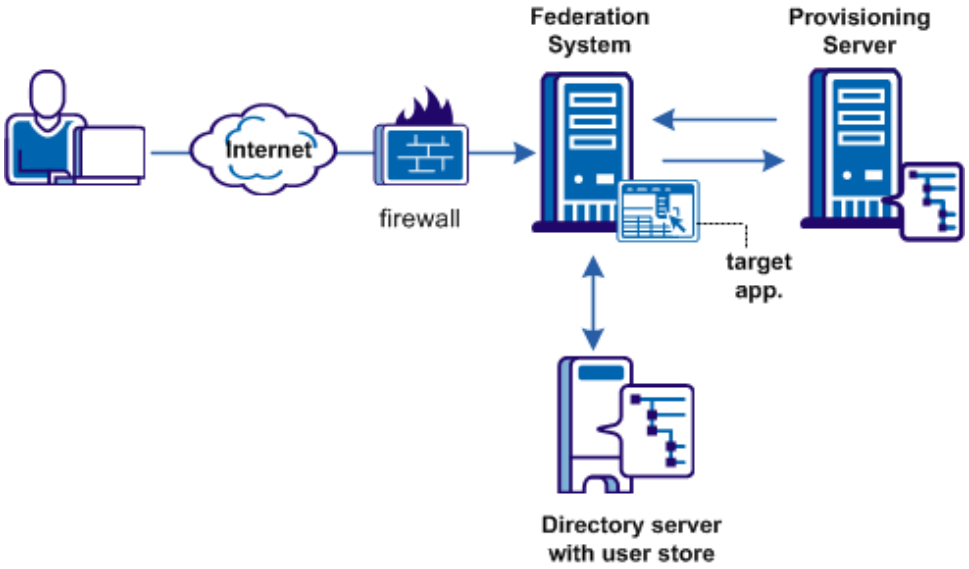
6. (Optional) Customize the linkaccount.jsp file to provide a custom user experience when the user is redirected after a failed authentication attempt. This file must POST the accountlinking and samlresponse parameters back to the Assertion Consumer Service. The accountlinking parameter must be set to yes. The page is in *federation_install_dir*/secure-proxy/Tomcat/webapps/affwebservices/public.

7. Select the Confirm step in the Partnership wizard and click Finish to save your changes.

Remote Provisioning

Remote provisioning employs a third-party provisioning application to create a new user account. The application then passes the necessary information back to the CA SiteMinder® Federation Standalone system. The federation system uses the data to create a user credential.

The following figure shows how a remote provisioning setup can be configured.



The high-level provisioning process is as follows:

1. The Policy Server at the relying party receives a request for a resource along with an assertion. However, the user cannot be found in the user directory.
2. With provisioning enabled, the Policy Server processes an active response containing assertion data and generates a cookie with the assertion data. Additionally, a cookie that keeps state is generated to indicate a provisioning request is in place.
3. The browser is redirected with an open-format cookie or headers to a provisioning application.
4. The provisioning application typically prompts the user to log in. After the user logs in, the application reads the cookie or the headers. The application uses the assertion data and the login credentials to establish a user account.

   The provisioning application can consume the open-format cookie using the CA SiteMinder® Federation Standalone Java or .NET SDK.

5. The browser redirects the user back to the assertion consumer service at the relying party after an account has been provisioned. A cookie that maintains state information about provisioning is examined to verify that the user has been provisioned. A credential is created and passed to the authentication scheme.

   Note: The provisioning application must know the URI of the assertion consumer service at the relying party. For example, the SAML 2.0 URI for CA SiteMinder® as the relying party is https://*sp_server:port*/affwebservices/public/saml2assertionconsumer.

6. The Policy Server attempts user disambiguation a second time. Assuming that provisioning is successful, the user is authenticated and cookies or headers are sent to the target application.

   The redirect mode that you select for the target application determines the data delivery method to the target application.

7. The user is redirected to the target resource.

Delivery of Assertion Data to the Provisioning Application

To accomplish remote provisioning, CA SiteMinder® Federation Standalone redirects the browser with the assertion data to the provisioning application.

CA SiteMinder® Federation Standalone can pass the assertion data using one of three methods:

Legacy cookie

Delivers SAML assertion information in a legacy cookie generated by CA SiteMinder® Federation Standalone. The cookie contains a login ID based on the assertion data. If a legacy cookie is used, then the CA SiteMinder® Federation Standalone Java SDK must be installed on the system with the provisioning application so that the provisioning application can read the legacy cookie.

Note: If you use the legacy cookie, the CA SiteMinder® Federation Standalone system and the remote provisioning system must be in the same domain.

Open format cookie

Delivers SAML assertion information in an open format cookie. The cookie contains a login ID based on the assertion data.

Note: If you use the open format cookie, the CA SiteMinder® Federation Standalone system and the remote provisioning system must be in the same domain.

The cookie can be created in one of two ways:

- The CA SiteMinder® Federation Standalone SDK creates the cookie.

  If you select one of the FIPS algorithms (AES algorithms), you are required to use a CA SiteMinder® Federation Standalone SDK to generate the cookie. If you are planning to use the .NET SDK, you are required to use the AES128/CBC/PKCS5Padding encryption algorithm. If the provisioning application uses .NET then the CA SiteMinder® Federation Standalone .NET SDK can be installed on the provisioning server and used to read the open format cookie.

  The provisioning application must use the same language as the SDK that it is using to create a cookie. If you are using the CA SiteMinder® Federation Standalone Java SDK, the application must be in Java. If you are using the .NET SDK, the application must support .NET.

- You manually create an open format cookie.

  To create an open format cookie without using a CA SiteMinder® Federation Standalone SDK, use any programming language to create the cookie. Review the details about the contents of the open format cookie (see page 405).

  The language you use to write the cookie must support UTF-8 encoding and any of the PBE encryption algorithms that CA SiteMinder® Federation Standalone uses for password-based encryptions, which include:

  - PBE/SHA1/AES/CBC/PKCS12PBE-1000-128
  - PBE/SHA1/AES/CBC/PKCS12PBE-1000-192
  - PBE/SHA1/AES/CBC/PKCS12PBE-1000-256
  - PBE/SHA256/AES/CBC/PKCS12PBE-1000-128

- PBE/SHA256/AES/CBC/PKCS12PBE-1000-192
- PBE/SHA256/AES/CBC/PKCS12PBE-1000-256
- PBE/SHA1/3DES_EDE/CBC/PKCS12PBE-1000-3
- PBE/SHA256/3DES_EDE/CBC/PKCS12PBE-1000-3

The provisioning application cannot read the open format cookie without an SDK if you select FIPS-compatible (AES) algorithm to encrypt the cookie.

You must also ensure that the open format cookie gets set in the user's browser.

Note: If you installed CA SiteMinder® Federation Standalone in FIPS-only mode, only the open format cookie is available.

HTTP headers

If proxy mode is used, this information can also be passed as HTTP headers. If you use HTTP headers, the CA SiteMinder® Federation Standalone system and the remote provisioning system can be in different domains.

The delivery option is configurable in the Application Integration step of the Partnership wizard.

After the user is redirected to the provisioning application, CA SiteMinder® Federation Standalone no longer has control over the process. If provisioning a user account is a time-consuming process, the provisioning application is responsible for handling this situation, for example, by sending a message to the user that provisioning is in process. This information lets the user know not to keep trying to log in before an user account is available.

Remote Provisioning Configuration

Configuring remote provisioning requires that you determine a delivery option of the assertion data and supply the URL of the provisioning server.

In addition to configuring remote provisioning, you can select the Allow IdP to create User Identifier option. This option enables the IdP to create a persistent identifier if no identifier for the user exists. This Allow/Create feature is not exclusively for provisioning using local account linking, though it is required for the local method.

You can enable the Allow/Create feature together with remote provisioning, if you want the IdP to generate a user identifier that is sent with other attributes to the remote provisioning server. The application at the remote provisioning server determines how it uses the generated identifier. The application can perform local account linking; however, this is not CA SiteMinder® Federation Standalone local account linking.

To configure remote provisioning

1. Begin at the Application Integration step of the Partnership wizard.
2. Select the provisioning type in the User Provisioning group box.

   Note: Click Help for a description of fields, controls, and their respective requirements.
3. If you select Remote as the provisioning type, complete the following additional fields:
   - Delivery Option
   - Provisioning Server URL
4. If you specify open format cookie as the delivery option, you must complete the additional settings in the Open-format Cookie group box.

   These settings include the name of the cookie, the algorithm that encrypts the cookie and the encryption password. Optionally, you can enable an HMAC function to verify the integrity of the cookie.
5. Select the Confirm step in the wizard and click Finish to save your changes.

You have completed remote provisioning configuration.

More information:

Delivery of Assertion Data to the Provisioning Application (see page 313)

Failed Authentication Handling Using Redirect URLs (Relying Party)

Assertion-based authentication can fail at the site that consumes assertions. If authentication does fail, you can configure CA SiteMinder® Federation Standalone to redirect the user to different applications (URLs) for further processing. For example, when user disambiguation fails, CA SiteMinder® Federation Standalone can be configured to send the user to a provisioning system, which could create a user account based on the information found in the SAML assertion.

Setting up redirect URLs is optional and is only configurable at the relying party.

To configure the redirect URLs

1. Begin at the Application Integration step of the Partnership wizard.
2. In the Status Redirect URL section of the dialog, configure only those settings for the failure conditions for which you want to redirect users. The settings in the Status Redirect URLs group box are:
   - User Not Found
   - Invalid SSO Message
   - Unaccepted User Credential (SSO message)

   Note: Click Help for a description of fields, controls, and their respective requirements.
3. For each redirect option you configure, specify the method by which CA SiteMinder® Federation Standalone redirects the user. The options are:

   302 No Data (default)

   Redirects the user with an HTTP 302 redirect and no data.

   HTTP Post

   Redirects the user with the HTTP Post protocol.

Configuration of the redirect URLs is complete.

Chapter 23: Export Metadata to Aid Partnership Configuration

This section contains the following topics:

Metadata Export Overview (see page 319)
Entity-level Metadata Export (see page 320)
Partnership-Level Metadata Export (see page 320)
How To Enable WS-Federation Metadata Exchange (see page 321)

Metadata Export Overview

A local entity generates metadata to help a remote entity create its entities and form partnerships. Metadata makes the partnership configuration more efficient because many aspects of the partnership are defined in the metadata file. A remote partner can import metadata and can create a partnership or a remote entity that is based on the information in a metadata document.

You can export metadata from an existing local asserting or relying entity.

The Administrative UI offers several options for exporting metadata:

- Export from a local entity.
- Export from a local partnership.
- Metadata exchange for local WSFED partnerships.

Regardless of whether you send metadata using a file or using the metadata exchange profile, the end goal of acquiring metadata is the same.

Note: For SAML 1.1, the terms in a metadata file are SAML 2.0 terms. This convention adheres to the SAML specification. When you import the SAML 1.1 data, the terms are imported correctly using SAML 1.1 terminology.

Entity-level Metadata Export

You can export data from a local entity. When you export metadata at the entity level, provide a partnership name for the data you are exporting. The export at this level defines basic partnership data.

Follow these steps:

1. Log in to the Administrative UI
2. Click Federation, Entities.
3. Click the Action pull-down menu next to any local entity in the list and select Export Metadata.

   The Export Metadata dialog opens.
4. Specify a new partnership name. The metadata file that results from the export contains information to establish a basic partnership.
5. Complete the remaining fields on the dialog. Be sure to fill in the settings in the Metadata Export Options section of the dialog.

   Note: Click Help for a description of fields, controls, and their respective requirements.
6. Click Export.
7. A dialog prompting you to open or save the metadata file displays.

   Only open it to view it.
8. Save the data to an XML file on your local system.

The metadata is exported to the specified XML file. You can send this file to any partner.

Partnership-Level Metadata Export

You can export data from a local partnership. The export at this level defines basic partnership data.

Follow these steps:

1. Log in to the Administrative UI
2. Click Federation, Partnerships.
3. Select the Action pull-down menu next to any partnership in the list.
4. Select Export Metadata.

   The Export Metadata dialog opens.

5. Review the information. The metadata file that results from the export contains information to establish a basic partnership.
6. Complete the settings in the Metadata Export Options section for signing the metadata document and validating it.

   Note: Click Help for a description of fields, controls, and their respective requirements.
7. Click Export.
8. A dialog prompting you to open or save the metadata file displays.

   Only open it to view it.
9. Save the data to an XML file on your local system.

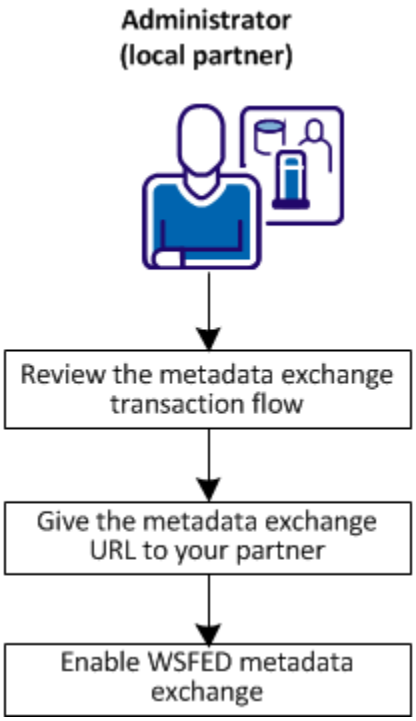
The metadata is exported to the specified XML file. You can send this file to any partner.

How To Enable WS-Federation Metadata Exchange

The Policy Server supports the Web Services Metadata Exchange profile for WS-Federation partnerships. This web service enables the CA SiteMinder® local partner to respond to requests from a remote partner for metadata. The exchange occurs as an HTTP request and response.

The use of the HTTP protocol lets a remote entity configure the federation programmatically. An application can use the URL to gather the necessary information.

The following graphic shows the configuration steps for metadata exchange.



Complete the following configuration for metadata exchange:

1. Review the metadata exchange transaction flow. (see page 322)
2. Give the metadata exchange URL to your partner. (see page 323)
3. Enable WSFED metadata exchange (see page 323).

Metadata Exchange Transaction Flow

A metadata exchange transaction has the following process flow:

1. A remote partner sends a request to the metadata exchange URL provided by the local partner.
2. The local partner sends the metadata back in an HTTP response to the remote partner. The Policy Server secures the metadata by signing the response. The certificate that lets the remote partner verify the response is in the response.

   The Policy Server generates the metadata document at the time of the request. This document is not stored at the local partner.
3. The remote partner verifies the signature of the response. Assuming the signature is valid, it parses the metadata document and uses the information to establish entities and partnerships.

Give the Metadata Exchange URL to Your Partner

Before any metadata transaction occurs, give the URL for metadata exchange requests to your remote partners. A federated partner must send the request to the following URL:

https://*server:port*/affwebservices/public/FederationMetadata/*partnership_name*

server:port

Name of the system hosting the metadata exchange service.

partnership_name

Name of a configured partnership.

Enable WSFED Metadata Exchange

Enable the metadata exchange feature at a local WS-Federation partner.

Follow these steps:

1. Log in to the Administrative UI.
2. Select the WSFED partnership that you want to modify.
3. In the Configure Partnership step of the partnership wizard, select the Enable Metadata Exchange check box.
4. Navigate to the Confirm step and click Finish.
5. Return to the main Partnership Federation tab (Federation, Partnership Federation).
6. Select Metadata Exchange Configuration in the left pane.

   The Metadata Exchange Configuration screen displays.
7. Provide the values to sign the response.
8. Click Save.

Metadata exchange is now configured for the partnership.

Chapter 24: Failover Support for Federation System

This section contains the following topics:

Failover Introduction (see page 325)
How to Configure Failover (see page 327)
How to Configure Failover with SSL Enabled (see page 329)
Maintain the Same Configuration for Each System (see page 334)

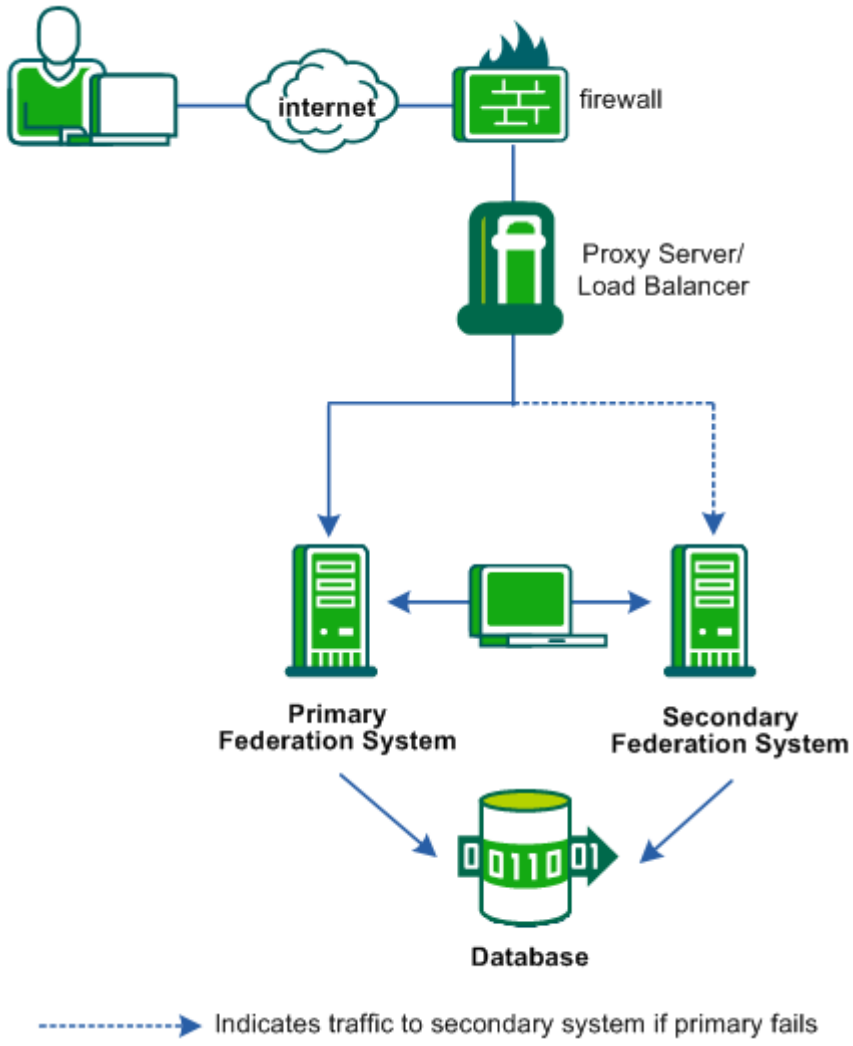
Failover Introduction

Failover support ensures that CA SiteMinder® Federation Standalone is not a single point of failure in your federated network. Failover builds redundancy into your network by configuring a primary and secondary CA SiteMinder® Federation Standalone system. If the primary CA SiteMinder® Federation Standalone system fails, the back-up system can perform the necessary federated communication.

Failover can be configured for CA SiteMinder® Federation Standalone acting as the asserting party and the relying party.

Note: If you enabled the CA SiteMinder® Connector, failover support is available for the Connector registration process. Instructions are described in the section Configure the CA SiteMinder® Connector.

The following figure shows a CA SiteMinder® Federation Standalone deployment with failover. If the primary system fails, transactions are directed to the secondary system.



As shown in the previous figure, CA SiteMinder® Federation Standalone is installed on two machines that use the same database.

How to Configure Failover

Configuring failover requires the following tasks:

- Installing CA SiteMinder® Federation Standalone on at least two systems.
- Running the Configuration wizard on the two federation systems.
- Setting up a proxy server or load balancer that will manage failover of the federation systems.

We recommend that you configure each federation system before configuring failover for the proxy server or load balancer.

Important! If you plan to use SSL for federation services, follow the instructions for an SSL-enabled failover environment (see page 329).

Set up Failover at Each Federation System

To enable failover in a federation deployment, a primary and a secondary CA SiteMinder® Federation Standalone system must be installed and configured.

For SSL-enabled failover environments, follow the instructions to enable SSL for a failover environment (see page 329).

Important! For Solaris platforms, treat Solaris zones as phyical machines. Install and configure separate CA SiteMinder® Federation Standalone instances in each zone. CA SiteMinder® Federation Standalone does not support failover from one zone to another for a single instance because the zones have different Host IDs.

Follow these steps:

1. Install the product on each system, specifying the same Federation Administrator Password for each installation.

   Note: The product can run in standalone or proxy mode, but the primary and secondary server must use the same mode.

2. Run the federation system Configuration wizard on each system using the same database information for both systems.

3. Log in to the Administrative UI.

4. From the Infrastructure tab, select System Settings.

5. Change the Global Base URL to include the host and port of the proxy server or load balancer in your federated network. Setting this URL helps ensure that the default URL for all entities in any partnership is correct.

   If CA SiteMinder® Federation Standalone uses more than one virtual host or domain, modify the server.conf file to include all entries.

   To modify the server.conf file

   a. Navigate to *federation_install_dir*/secure-proxy/proxy-engine/conf.

   b. Open the server.conf file in an editor.

   c. Go to the # Default Virtual Host section.

   d. Add the base URL to the **hostnames** setting using fully qualified host names, as follows:

      <VirtualHost name="default">

      hostnames="defaultbaseurl.example.com:80, newbaseurl.example.com:80"

      </VirtualHost>

      Note: Specify multiple *host_name:port* entries for the hostnames setting, separating each entry with a comma.

      Example:

      <VirtualHost name="default"

      hostnames=lb5.example.com:80

      </VirtualHost>

Both CA SiteMinder® Federation Standalone systems are pointing to the same database. A proxy server or load balancer can be set up to failover from the primary system to the secondary.

Set up the Proxy Server or Load Balancer for Failover

You can direct a proxy server or load balancer to failover to CA SiteMinder® Federation Standalone.

Note: The administrator of the proxy server or load balancer must know how to set up failover for the system in the deployment.

Follow these steps:

1. Identify one CA SiteMinder® Federation Standalone system as the primary host and the other as the secondary host.

   Do not configure load balancing for the systems.

2. Configure the proxy server or load balancer for the CA SiteMinder® Federation Standalone deployment, making sure to pass the following URLs to the CA SiteMinder® Federation Standalone systems:

   - /affwebservices/*
   - /siteminderagent/*

   These URLs enable the proxy server or load balancer to balance traffic between the CA SiteMinder® Federation Standalone systems.

The proxy server or load balancer is now configured.

How to Configure Failover with SSL Enabled

You can enable SSL in a failover environment whether your federation system is sitting behind a load balancer or a proxy server. There are specific configuration instructions for this type of setup.

Configuring SSL-enabled failover requires the following tasks:

- Installing CA SiteMinder® Federation Standalone on at least two systems.
- Running the Configuration wizard on the two federation systems.
- Enabling SSL for the embedded Apache Web Server (only if the federation system is behind a load balancer).
- Migrating the SSL configuration from the primary to the secondary system (only if the federation system is behind a load balancer).
- Setting up a proxy server or load balancer that will manage failover of the federation systems.

We recommend that you configure each federation system before configuring failover for the proxy server or load balancer.

Configure SSL-enabled Failover Behind a Load Balancer

You can configure the system to sit behind a TCP load-balancer. The load balancer passes the requests to the system, which then handles the server-side SSL processing.

Follow these steps:

1. Install the product on each system, specifying the same Federation Administrator Password for each installation.

   Note: The product can run in standalone or proxy mode, but the primary and secondary server must use the same mode.

2. Run the Configuration wizard and use the same database connection information for both systems.

3. The Configuration wizard prompts for the Apache Configuration information. Specify the same virtual host name in the Server Name setting for the primary and secondary federation systems. Both systems must use the same virtual host name.

   If the product is using more than one virtual host or domain, modify the server.conf file for the proxy engine. The server.conf file must list all host names and domains. Add the names to the hostnames field of the Default VirtualHost.

   To edit server.conf

   a. Navigate to the following directory:

      Windows: *federation_install_dir*\secure-proxy\proxy-engine\conf

      UNIX: *federation_install_dir*/secure-proxy/proxy-engine/conf

   b. Open the server.conf file in an editor.

   c. Go to the # Default Virtual Host section and add the names to the hostnames setting using a fully qualified URL, as follows.

      <VirtualHost name="default">

      hostnames="virtualhost1.example.com, virtualhost2.example.com"

      </VirtualHost>

      Note: You can specify multiple URLs for the hostnames setting, separating each entry with a comma.

4. Log in to the Administrative UI.

5. Click Infrastructure, System Settings.

6. Change the Global Base URL to include the host and port of the Proxy Server or load balancer in your federated network. Setting this URL helps ensure that the default URL for all entities in any partnership is correct.

   To modify the server.conf file

   a. Navigate to *federation_install_dir*/secure-proxy/proxy-engine/conf.

   b. Open the server.conf file in an editor.

   c. Go to the # Default Virtual Host section.

   d. Add the base URL to the **hostnames** setting using fully qualified host names, as follows:

      <VirtualHost name="default">

      hostnames="defaultbaseurl.example.com:80, newbaseurl.example.com:80"

      </VirtualHost>

   Note: Specify multiple *host_name:port* entries for the hostnames setting, separating each entry with a comma.

7. Enable SSL for the embedded Apache Web Server on the primary federation system.

8. Migrate the Apache SSL configuration to the secondary system in the failover deployment.

9. At the load balancer, configure multiple IP addresses for the same host name, which map to the federation system.

Migrate the SSL Setup to the Secondary System

After the Apache SSL is configured at the primary CA SiteMinder® Federation Standalone machine, it can be migrated to the secondary machine behind the load balancer.

Note: This procedure does not apply if CA SiteMinder® Federation Standalone is behind a proxy server.

Ensure that the following criteria is met:

- Same certificate is used for each CA SiteMinder® Federation Standalone machine.
- Each CA SiteMinder® Federation Standalone machine must be configured with the same host name.
- CA SiteMinder® Federation Standalone is accessed through a load balancer.
- All machines must be of the same platform (Windows/Solaris/Linux).

To copy the SSL configuration to the secondary machine

1. Enable Apache SSL on the primary CA SiteMinder® Federation Standalone machine. Once enabled, the following components are available:

   - SSL server cert

     federation_install_dir/secure-proxy/SSL/certs/server.crt

   - CA bundle

     federation_install_dir/secure-proxy/SSL/certs/ca-bundle.cert

   - SSL server key

     federation_install_dir/secure-proxy/SSL/keys/server.key

   - certificate request file

     federation_install_dir/secure-proxy/SSL/keys/fedmgrsslcertrequest.pem

   - SSL properties file

     federation_install_dir/config/fedmanager.properties

2. Import the CA certificate that signed the SSL Server Certificate to the secondary machine. Use the Administrative UI to import the certificate.

   This certificate should be imported before or during the SSL configuration process on the primary machine. It is recommended that you use the same alias as was used for this certificate on the primary machine.

3. Copy each of the files listed in step 1 to the same locations on the secondary machine. The folders should already exist.

   Note the following:

   - The secondary machine should already have a copy of ca-bundle.cert. That copy should be backed up or deleted; the new copy from the primary machine has additional data that the secondary machine requires.
   - Copying the certificate request file (fedmgrsslcertrequest.pem) is only required if you want to retrieve it using the Administrative UI on the secondary machine. If not, do not copy the file.
   - The SSL properties file should contain at least the following two properties:
     - fedmgr.ssl.enabled, set to Y.
     - fedmgr.ssl.ca.alias, set to the alias of the CA that signed the SSL server certificate request.
     - If you used a different alias when importing this certificate on the secondary machine, update this property with the alias value you actually used.

The configuration is now migrated and you can activate SSL on the secondary system.

Activate SSL on the Secondary Failover System

After migrating the Apache SSL configuration to the secondary system, enable SSL.

To activate SSL on the secondary machine (Windows)

1. Open a command prompt window on the secondary machine.
2. Navigate to the *federation_install_dir*/secure-proxy/httpd/bin folder.
3. Execute the following command:

   configssl.bat -enable
4. Stop and restart the CA SiteMinder® Federation Standalone services using the following shortcuts:

   If you logged in as a network user and not a local administrator, right-click the shortcut and select Run as administrator.

   - Start, All Programs, CA, Federation Standalone, Stop services
   - Start, All Programs, CA, Federation Standalone, Start services

To activate SSL on the secondary machine (UNIX)

1. Navigate to the *federation_install_dir*.
2. Shut down CA SiteMinder® Federation Standalone services by executing the command:

   ./fedmanager.sh stop
3. Restart CA SiteMinder® Federation Standalone services in SSL-enabled mode by executing the command:

   ./fedmanager.sh startssl
4. When prompted, enter the Administrative UI password to enable startup of the Apache web server in SSL mode.

Configure SSL-enabled Failover Behind a Proxy Server

If CA SiteMinder® Federation Standalone is behind a proxy server, the proxy server handles the SSL processing. CA SiteMinder® Federation Standalone cannot process SSL because many proxy servers cannot delegate the processing of SSL requests to other systems. Consequently, configure the proxy server with the server certificate and the CA certificates that signed the server certificate and any remote client certificates.

There is no specific SSL configuration required on the CA SiteMinder® Federation Standalone machines sitting behind the proxy server.

Set up the Proxy Server or Load Balancer for Failover

You can direct a proxy server or load balancer to failover to CA SiteMinder® Federation Standalone.

Note: The administrator of the proxy server or load balancer must know how to set up failover for the system in the deployment.

Follow these steps:

1. Identify one CA SiteMinder® Federation Standalone system as the primary host and the other as the secondary host.

   Do not configure load balancing for the systems.

2. Configure the proxy server or load balancer for the CA SiteMinder® Federation Standalone deployment, making sure to pass the following URLs to the CA SiteMinder® Federation Standalone systems:

   - /affwebservices/*
   - /siteminderagent/*

   These URLs enable the proxy server or load balancer to balance traffic between the CA SiteMinder® Federation Standalone systems.

The proxy server or load balancer is now configured.

Maintain the Same Configuration for Each System

If you make any changes to the configuration, always administer these changes to the primary CA SiteMinder® Federation Standalone system then export the configuration to the secondary machine.

Note the following information in regard to configuration changes:

Delay After Configuration Changes

Changes that you make using the primary system UI are not always immediately available to the secondary system. UI-managed data that is stored in the database is available to the secondary system because the primary and secondary systems share the same database. However, there can be a delay before the policy engine of the secondary system picks up the changes.

Some Configuration Changes Require Restart

Some configuration changes require a restart of the system. If you change the primary system configuration and this change requires a restart, restart the secondary system also.

Perform Administration on the Same Primary System

Always perform administration on the same primary system. You can disable UI administration on the secondary machine to enforce this practice.

Follow these steps:

1. Log on to the Administrative UI.
2. Select Infrastructure, System Settings.

   The Configure System Settings dialog displays.
3. Click Disable Administration in the UI Settings group box.

   A message appears asking you to confirm the action.
4. Click Yes to disable UI administration.

   An Administration Disabled dialog displays while all other parts of the UI become unavailable. This dialog lets you reenable administration.

Chapter 25: Load Balancing Support for Federation System

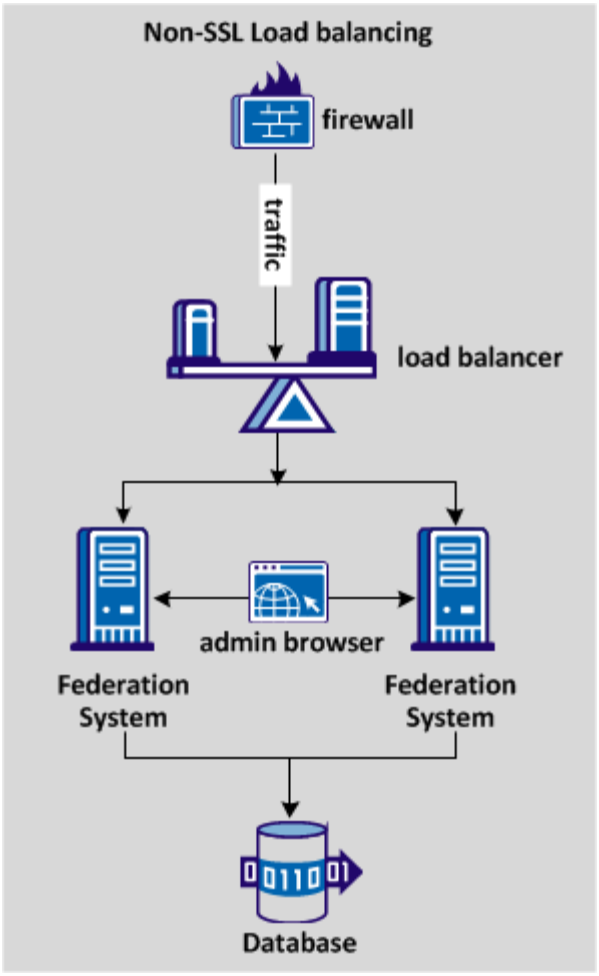
This section contains the following topics:

How to Configure Load Balancing (see page 337)

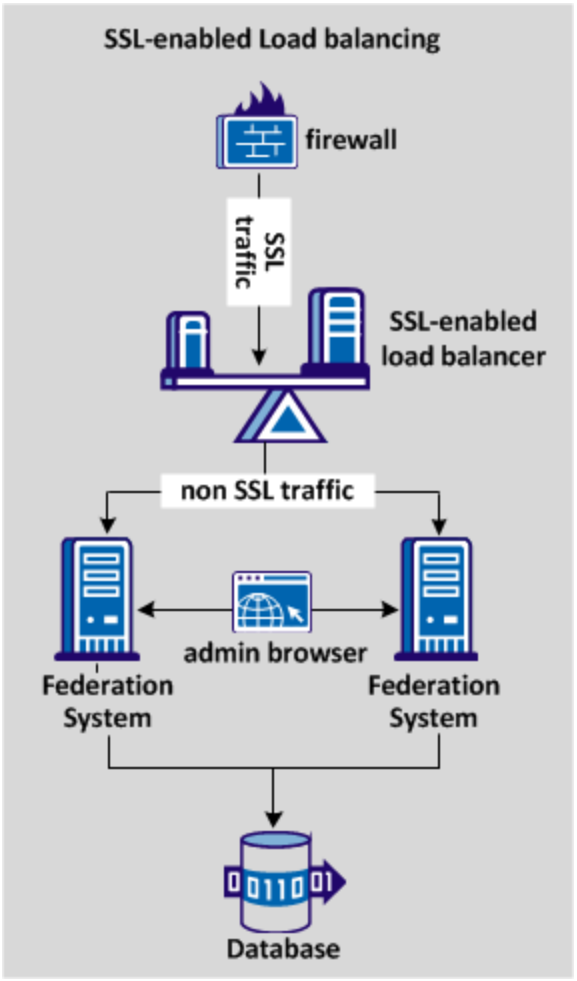
How to Configure Load Balancing

Load balancing distributes communication activity evenly across a network so that no single device is over burdened. CA SiteMinder® Federation Standalone is intended as a standalone endpoint and has no built-in support for load balancing. However, it is possible to enable a simple load-balancing deployment in a network using CA SiteMinder® Federation Standalone. SSL-enabled load balancing is optional, but it is recommended for sending sensitive data.

The following diagram shows a non-SSL load-balancing deployment.



The following diagram shows an SSL load-balancing deployment, distributing traffic across two systems that use the same database store.



Complete the following configuration steps for load balancing traffic:

1. Configure the load balancer.
2. Set up two or more CA SiteMinder® Federation Standalone systems to work with a load balancer.
3. (Optional) If the load balancer is using SSL, configure CA SiteMinder® Federation Standalone to handle SSL redirection.

Configure the Load Balancer

Configure the load balancer to work in a network with CA SiteMinder® Federation Standalone. Use the load balancer host and port to redirect a user who requests a resource that resides on the CA SiteMinder® Federation Standalone system. Use of the load balancer host and port applies to all resources on the CA SiteMinder® Federation Standalone system.

Note: This procedure assumes that the load balancer administrator knows how to set up the system in the deployment.

Follow these steps:

1. Configure the load balancer to map IP addresses and host names for the federation deployment.
2. Configure the load balancer for the deployment, making sure to pass the following URLs through to the CA SiteMinder® Federation Standalone systems:
   - /affwebservices/*
   - /siteminderagent/*
   - /forms/*

   These URLs enable the load balancer to balance traffic between the federation systems.
3. (Optional) Configure the load balancer to handle SSL traffic.

   If the load balancer is SSL-enabled, all federation traffic comes in to the load balancer over SSL. However, the load balancer sends the traffic to the CA SiteMinder® Federation Standalone systems over a non-SSL (HTTP) connection.

The load balancer is now configured to work with CA SiteMinder® Federation Standalone systems.

Set up the Federation Systems to Work with a Load Balancer

To use load balancing across a federation deployment, set up two or more CA SiteMinder® Federation Standalone systems.

Note: The procedure assumes that all systems are version r12.52.

Follow these steps:

1. Install the product on each system, specifying the same Federation Administrator Password for each installation.

   Note: Whether the prodcut is run in standalone or proxy mode, the servers must use the same mode.

2. Run the Configuration Wizard on one system.
3. Log in to the Administrative UI.
4. Navigate to Infrastructure, System Settings.
5. In the Server Settings section, change the Global Base URL to include the host and port of the load balancer in your network. Set this URL so that the default URL for all partnership entities is correct.
6. Set up a federation partnership by completing the following tasks
   a. Import certificates and private keys.
   b. Establish user directories connections (see page 73).
   c. Configure local entities (see page 107).
   d. Specify a remote entity (see page 107).
   e. Configure a partnership between the local and remote entities.
   f. Verify that the federation works with the remote partner.
7. Run the Configuration Wizard on the secondary systems, using the same virtual host name of the load balancer that you entered for the first system.

   Each federation system must use the same virtual host name. The virtual host name is the host that you specify for the Server Name in the Apache configuration when you run the Configuration Wizard.

   If the product uses more than one virtual host or domain, modify the server.conf file to include the additional entries.

   To modify the server.conf file

   a. Navigate to *federation_install_dir*/secure-proxy/proxy-engine/conf.
   b. Open the server.conf file in an editor.

c. Go to the # Default Virtual Host section.

d. Add the base URL to the **hostnames** setting using fully qualified host names, as follows:

```
<VirtualHost name="default">
hostnames="defaultbaseurl.example.com:80, newbaseurl.example.com:80"
</VirtualHost>
```

Note: Specify multiple *host_name:port* entries for the hostnames setting, separating each entry with a comma.

Example:

```
<VirtualHost name="default"
    hostnames=lb5.example.com:80
</VirtualHost>
```

8. Migrate SSL keys and certificates that are stored by the embedded Apache and Tomcat web servers.

   - Follow the SSL migration procedure (see page 367) to complete this task. Migrating SSL data lets you avoid the purchase of a new key or certificate.
   - Generate a new key/certificate request and then get the certificate signed. SSL certificates are not included in the imported configuration file.

   Note: Replicate any change to the certificate configuration on one system to all other systems. Make configuration changes from the Certs and Keys page in the UI. Changes include adding or removing certificates, keys, or CRL data.

9. Log in to the Administrative UI on the other systems that do not have partnerships configured.

10. Navigate to Infrastructure, System Settings. In the UI Settings section, click Disable Administration.

    Access the Administrative UI locally, without going through the load balancer. If the other systems are up and running, enable administration on only one system. If the administration system is disabled at any time, log in a different system and reenable administration.

Now that all federation systems are pointing to the same data store, the configured load balancer is able to balance traffic between the systems.

Configure Redirections to an SSL Load Balancer (optional)

If the load balancer uses SSL, we recommend that you configure the system to redirect traffic over an SSL connection. To redirect traffic, modify the following two files on each federation system:

- LocalConfig.conf
- httpd.conf

Note: Modify these files on all federation systems that are redirecting traffic.

Follow these steps:

1. Navigate to *federation_install_dir*/secure-proxy/proxy-engine/conf/defaultagent.
2. Open the WebAgent.conf file in an editor. Uncomment the line that begins **localconfigfile** then save the file.
3. Open the LocalConfig.conf file in an editor.
4. Add the following settings to the LocalConfig.conf file then save the file:

   `HttpsPorts="443"`

   Specify the port on which the load balancer is listening.

   `GetPortFromHeaders="YES"`
5. Navigate to *federation_install_dir*/secure-proxy/httpd/conf.
6. Open the httpd.conf file in an editor.
7. Locate the SeverName setting and specify the load balancer *hostname:port*. Do not enter the federation system server host name.

   Example:

   `ServerName lb5.example.com:443`
8. After the ServerName setting, add the UseCanonicalName setting and set it to On. Example:

   `UseCanonicalName on`

Federation system now redirects traffic over an SSL connection.

Chapter 26: Federation System Administration

This section contains the following topics:

Server Status Monitoring (see page 345)
Modify the System Settings (see page 345)
Deployment Settings (see page 346)
How to Configure Federation System Administrators (see page 351)
Administrator Session Management (see page 355)

Server Status Monitoring

The Server Status dialog provides a snapshot of information that can be useful for checking the condition of the server. Examples include enhancing system performance or verifying that the installation went as expected.

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

To view the server settings

1. Log in to the Administrative UI.
2. Select Infrastructure, Server Status.
3. Review the information on the status page.

You can click Refresh at any time to see updated server information.

Modify the System Settings

The Configure System Settings dialog lets you specify the number of active server threads and the number of active server connections allowed, which may have an impact on system performance. It also allows you to disable and re-enable UI Administration for the local host.

Follow these steps:

1. Start the Administrative UI.
2. From the Infrastructure tab, select System Settings.

3. Modify any of the settings, as necessary.

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

4. Click Save.
5. Restart CA SiteMinder® Federation Standalone.

   Your changes take effect after the server is restarted with the exception of the Disable Administration feature, which takes effect immediately.

Deployment Settings

The Deployment Settings let you do the following:

- View the FIPS mode setting.
- View the CA SiteMinder® Federation Standalone deployment mode (standalone or proxy).
- Specify valid federation domains.
- Configure the CA SiteMinder® Connector if CA SiteMinder® Federation Standalone is operating with the CA SiteMinder® Connector enabled.
- Modify the CA SiteMinder® Federation Standalone session cookie names.

Deployment Modes and FIPS Settings

The Deployment Settings shows the status of the deployment mode and FIPS mode that you selected when installing and configuring CA SiteMinder® Federation Standalone. Additionally, you can specify valid federation domains, and can set a prefix to protect HTTP headers in proxy mode.

The process for modifying each setting differs.

Valid Federation Domains and HTTP Header Prefix

Modify the Valid Federation Domains and HTTP Header Prefix entries from the UI. Changing these settings is optional.

Follow these steps:

1. Enter values for the fields, if necessary.
2. Click Save in the right corner of the section.

FIPS Mode Changes

To change the FIPS Mode, run the Installation wizard again and choose a new setting.

Important! Anytime you change the FIPS mode, restart CA SiteMinder® Federation Standalone.

Deployment Mode Changes

To change the Deployment Mode, run the Configuration wizard again and change the mode.

More information:

Encryption and Decryption Algorithms (see page 409)

HTTP Header Protection for a Proxy Mode Deployment at the Relying Party

In a proxy mode deployment at the relying party, CA SiteMinder® Federation Standalone passes identity attributes from the SAML assertion to backend applications using HTTP headers. In most cases, the headers are secure. However, if an unauthorized user knows an assertion attribute name they can set this name as a header in a browser and gain access to the target application. The target application sees an expected header value and grants access to the resource without CA SiteMinder® Federation Standalone consuming an assertion.

By specifying a value for the HTTP Header Prefix setting, you can protect against the following scenario:

1. An unauthorized user learns the names of HTTP headers. These header names include prefixes.
2. The malicious user sends an incoming request, including the headers, to CA SiteMinder® Federation Standalone.
3. CA SiteMinder® Federation Standalone recognizes that the headers containing prefixes come from an incoming request and are not generated internally so it removes these headers.
4. Before CA SiteMinder® Federation Standalone passes its own legitimate headers to the target application, it adds the specified prefix to each header and passes the headers to the target application.

To set the HTTP Header Prefix

1. Navigate to Infrastructure, Deployment Settings.
2. Enter any valid string as a prefix in the HTTP Header Prefix field.

   You only see this field if you enabled Proxy Mode when installing CA SiteMinder® Federation Standalone.
3. Save your changes.

CA SiteMinder® Connector Settings

The CA SiteMinder® Connector lets CA SiteMinder® Federation Standalone integrate with a CA SiteMinder® environment for federated communication.

At the asserting party, the CA SiteMinder® Connector can work with CA SiteMinder® as a third-party WAM for delegated authentication. At the relying party, CA SiteMinder® can protect the server where the target resources reside. If CA SiteMinder® is performing access control, the CA SiteMinder® Connector contacts the Policy Server to establish a CA SiteMinder® session so that CA SiteMinder® grants the user access to the target resource.

For CA SiteMinder® Federation Standalone to operate with CA SiteMinder®, configure the CA SiteMinder® Connector settings in the Administrative UI.

All partnerships that use the CA SiteMinder® Connector use a single configuration and connect to a single CA SiteMinder® environment. Define the Connector configuration in the Deployment Settings of the Administrative UI. To enable the Connector for a given partnership, enable it at the partnership level. Disable the Connector at the partnership level or globally by disabling it in the Deployment Settings.

Important! If the Connector is disabled at the global level, CA SiteMinder® Federation Standalone ignores the check box at the partnership level.

Follow these steps:

1. Log in to the Administrative UI.
2. Select a partnership from the Federated Partnerships list.

   The Partnership dialog opens.
3. Do one of the following:
   a. At the relying party, navigate to the User Identification step in the Partnership wizard.
   b. At the asserting party, navigate to the Federation Users step in the Partnership wizard.

4. Select the Enable SiteMinder Connector check box.

   The configuration fields become available.

5. (Optional) Select the Enforce UserDN Comparison check box. Selecting this check box forces a comparison of the UserDN and UserDirectory Name entries between the user directory at CA SiteMinder® Federation Standalone and the directory at CA SiteMinder®.

   If you select this check box, the user directory for the CA SiteMinder® Federation Standalone deployment and the CA SiteMinder® deployment must be the same physical directory. The name for both of these directories must be the same for user store lookups. If you clear the check box, CA SiteMinder® Federation Standalone uses the Universal ID to find the user record so the directories do not have to be the same. If you rely on the Universal ID, each user must have a unique Universal ID. If the Universal IDs are not unique, the system accessing the user record can retrieve the wrong record.

6. Save your changes.

7. Navigate to the Infrastructure tab.

8. From the Infrastructure tab, select Deployment Settings.

   The Configure Deployment Settings dialog opens.

9. Fill in all the fields in the CA SiteMinder® Connector Settings section.

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

10. Select Register Host and provide the administrator credentials for the CA SiteMinder® Policy Server.

    This step registers CA SiteMinder® Federation Standalone as an Agent with the CA SiteMinder® Policy Server.

    Note: You can configure failover support for the host registration process by specifying more than one Policy Server. If the registration with the primary Policy Server fails, CA SiteMinder® Federation Standalone moves to the next Policy Server specified until the registration process completes successfully.

11. Select Save in the SiteMinder Connector Settings section of the dialog.

    Selecting Save in the CA SiteMinder® Connector Settings section is necessary after registering the host.

12. Restart the federation services according to your operating environment.

    - **Windows**

      Use the stop and start shortcuts as follows. If you logged in as a network user and not a local administrator, right-click the shortcut and select Run as administrator.

      a. Start, All Programs, CA, Federation Standalone, Stop services

      b. Start, All Programs, CA, Federation Standalone, Start services

- **UNIX**

  a. Open a command window.

  b. Run the following scripts:

  *federation_install_dir/*fedmanager.sh stop

  federation_install_dir/fedmanager.sh start

  Note: Do not stop and start the services as the root user.

The CA SiteMinder® Connector configuration is complete.

Cookie Settings for Session and Identity Cookies

CA SiteMinder® Federation Standalone supports single sign-on security zones. Single sign-on security zones provide configurable trust relationships between groups of applications within the same cookie domain.

Although single sign-on is enforced within the same zone, a user may be rechallenged when entering a different zone, depending on the trust relationship defined between the zones. Security zones included in a trusted relationship do not rechallenge a user that has a valid session in any zone in the group.

Security zone affiliation is reflected in cookie names. For CA SiteMinder® Federation Standalone, the default session and identity cookies are named FEDSESSION and FEDPROFILE.

Your federation partner possibly has an application that uses its own session or identity cookie. The names of the partner cookies can conflict with the names of CA SiteMinder® Federation Standalone's cookies. For example, if you are communicating with a CA SiteMinder® site, cookies named FEDSESSION and FEDPROFILE may exist because CA SiteMinder® generates its own session and identity cookies. In this case, you can change the global cookie zone prefix for CA SiteMinder® Federation Standalone so its cookies get renamed.

Note: If you have an application that is using a CA SiteMinder® Federation Standalone SDK, the values configured for the Global Cookie Zone and Encryption Password settings must match what the SDK uses. Be sure to share the values of these settings with the appropriate parties in your organization. At the asserting party, the SDK and web access management system need these values. At the relying party, CA SiteMinder® Federation Standalone and the target system that hosts the application need to know these values.

For additional information, see the CA SiteMinder® Federation Standalone *Java SDK Guide* or the *.NET SDK Guide.*

The other cookie parameters in this group box are the open format cookie settings. The open format cookie settings are used only for the open format cookie method of delegated authentication and apply on a global level not on a partnership basis.

Note: At the relying party, the configuration of this cookie data is done at the partnership level and not at a global level.

To change the cookie settings

1. Log in to the Administrative UI.
2. From the Infrastructure tab, select Deployment Settings.

   The Configure Deployment Settings dialog is displayed.
3. (Optional) Modify all the settings in the Cookie Settings section, as needed.

   Note: Click Help for a description of fields, controls, and their respective requirements.
4. Click Save in the right corner of the section.

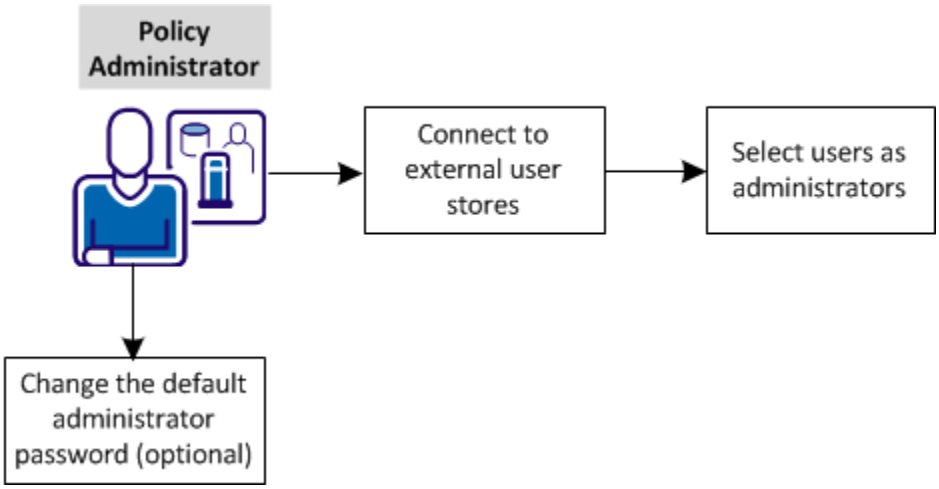
How to Configure Federation System Administrators

Several administrators in your company can be responsible for different aspects of federation management. Assign the administration of CA SiteMinder® Federation Standalone to multiple people in your organization to establish accountability and separation of responsibilities.

A default administrator account is always available to manage CA SiteMinder® Federation Standalone. After you add new administrators, optionally, disable the default administrator account.

Create and maintain new administrative users through the Administrative UI.

The following graphic shows the configuration tasks for configuring administrators:



Complete these tasks:

1. Connect to external user directories (see page 352).
2. Select users as administrators (see page 353).
3. Change the default administrator password (optional) (see page 354).

Connect to External User Stores

Create the connections to LDAP and ODBC external user stores. This step is required before you configure multiple administrators.

LDAP and ODBC are the two types of directories that the federation system supports.

Follow these steps:

1. Click the User Directory tab.
2. Click Connect to LDAP or ODBC.

   You can select Action, Modify to verify the configuration of an existing directory connection.
3. Configure any required settings in each section. Red dots mark the required parameters.

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

4. Enter a value for the Universal ID Attribute (LDAP) or Universal ID Column (OCBC). This value is required to configure multiple administrators.

   The universal ID value must be unique to identify individual users in a directory. For example, enter uid as a universal ID for an LDAP directory because each user has a uid. Do not use an attribute such as a job title because many users have the same title.

5. For LDAP directories only, specify values for the Start and End User DN Lookup fields. For example:

   Start User DN Lookup

   (uid=

   End User DN Lookup

)

6. Click Test Connection to verify that the connection is valid.

   You can click View Contents to list the contents of the user directory.

   Note:

   - For an LDAP directory connection, the View Contents button is displayed only if the Search Root, Start User DN Lookup, End User DN Lookup, and Universal ID Attribute values are set.
   - For an ODBC directory connection, the View Contents button is displayed only if the Universal ID Column value is set.

7. Click Save.

   If your settings are valid, you are redirected to the View User Directories dialog.

   The connection to the directory is configured.

Select Users as Administrators

After you establish connections to external user stores, select users to serve as administrators.

Follow these steps:

1. Log on to the Administrative UI.
2. Navigate to Infrastructure, Administrators.
3. Select Configure Administrative Authentication.
4. To complete these tasks, follow the configuration wizard:
   - Select an external user store.
   - Select one or more users as administrators.

- Determine the type of access privileges for each administrator. The options are:
  - Super User
  - Federation Administrator
  - Read Only User

  Note: Click Help for a description of each privilege.
- Decide whether to disable the default administrator. To prevent the use of a central administrator account, disable the default administrator. Without a default administrator, rely on individual administrators that you are able to audit.

5. Log out of the Administrative UI and wait several minutes for the changes to take effect.
6. Log back in to the Administrative UI with the credentials of a new administrator.
7. Return to the Administrators page to see the list of administrators is displayed.
8. (Optional). From the Action menu, modify or view an entry.

   You can change the privileges of the administrator and can enable/disable the administrator.

Multiple administrators are now available to divide ederation management tasks.

Change the Default Administrator Password (Optional)

For security reasons, change the password that gives the default administrator access to the Administrative UI. This task is optional.

Two methods are available to change the administrator password:

- Modify the password from the UI.
- Modify the password from a command line.

  If the administrator user account is locked or the administrator is not available, reset the password from the command line.

Change the Default Administrator Password from the UI

You can change the administrator password from the Administrative UI.

Follow these steps:

1. From the Infrastructure tab, select Password.

   The Change Administrator Password dialog displays.

2. Complete the fields for the old and new passwords.
3. Click Submit.
4. Restart the system.

The default administrator password is changed and active.

Change the Default Administrator Password from the Command Line

If the administrator user account is locked or unavailable, use the XPSConfig utility to change the administrator password from the command line.

Follow these steps:

1. On your federation system, open a command window.
2. Enter XPSConfig.

   On UNIX platforms, enter the name of the utility as shown here. The name is case-sensitive.

   The Products Menu displays.
3. Enter FED to select the federation product.
4. Enter 1.

   Option 1 is the option for the password.
5. Enter C to change the value of the password
6. Enter the new value at the prompt.
7. Enter Q to save and quit.

The new password is active.

Administrator Session Management

Only one administrative session can be active at a time. The single administrative session prevents simultaneous editing of federation objects if an administrator tries to establish a new session. When any new login attempt is made after an administrative session is established, you receive a warning message.

The warning message tells the administrator that a session exists. If the administrator proceeds to log in using the same credentials, the system invalidates the existing session and any unsaved data is lost. After the first session becomes invalid, the administrator of the first session is logged out. If the administrator tries any configuration activity, the system redirects the administrator to the login dialog.

The next section describes what happens when one administrator attempts to log in using the same credentials as an already established administrator session.

Administrative Session Interaction

The following scenarios result in an administrative session conflict:

Attempted Log in with Same Credentials

An administrator logs in to CA SiteMinder® Federation Standalone. A second administrator or the same administrator tries to log in using the same credentials as the first log in but from another browser session.

CA SiteMinder® Federation Standalone presents the warning dialog but the second user decides to log in, CA SiteMinder® Federation Standalone invalidates the first session. If the administrator of the first session tries to modify an object, CA SiteMinder® Federation Standalone alerts the administrator that a new session has invalidated the existing session and CA SiteMinder® Federation Standalone logs out the first administrator.

The second user can choose not to log in

Browser Session Closes but Administrator Not Logged out

An administrator logs in to CA SiteMinder® Federation Standalone. The administrator closes the browser session without logging out or the browser session closes unexpectedly and the administrator is not given a chance to log out. Another administrator logs in using the same credentials as the first administrator but from another browser session.

CA SiteMinder® Federation Standalone presents the warning dialog but the second user decides to log in, invalidating the first session.

If the administrator of the first session tries to resume the browser session and modify an object, CA SiteMinder® Federation Standalone alerts the administrator that a new session has invalidated his existing session. The first session was invalidated when the browser closed.

Note: Not all browsers permit a user to resume an unexpectedly closed browser session. For those browsers, CA SiteMinder® Federation Standalone does not present an alert that the existing session is invalid.

Administration is Disabled

An administrator logs in to CA SiteMinder® Federation Standalone. From the System Settings, the administrator disables administration. Another administrator tries to log in using the same credentials as the first administrator. CA SiteMinder® Federation Standalone presents the warning dialog but the second user decides to log in. CA SiteMinder® Federation Standalone invalidates the first session.

The first administrator, who did not log out or close the browser, tries to re-enable administration. CA SiteMinder® Federation Standalone displays a message telling the first administrator the session is invalid and logs the administrator out.

Disable UI Administration

The UI Settings group box lets you disable and re-enable CA SiteMinder® Federation Standalone administration on the local host.

The ability to disable administration of the UI is useful if two CA SiteMinder® Federation Standalone systems are set up to support failover. Administration of CA SiteMinder® Federation Standalone can only take place on the primary CA SiteMinder® Federation Standalone system. The configuration can then be exported to the secondary CA SiteMinder® Federation Standalone system.

Use the disable UI feature to disable administration on the secondary system so administration can only take place on the primary system.

Follow these steps:

1. Log in to the Administrative UI.
2. Navigate to Infrastructure, System Settings.
3. Click Disable Administration in the UI Settings group box.

   Disabling administration prevents any administrative actions.

Important! The change takes effect immediately after you confirm the action.

After you disable administration, an Administration Disabled dialog displays and all other parts of the UI become unavailable. Subsequent login attempts only display a warning message and a button to re-enable administration.

To re-enable administration

Click Enable Administration in the Administration Disabled dialog.

All parts of the UI become active again.

Chapter 27: SSL Administration for Federation System

This section contains the following topics:

SSL Administration for the Apache Web Server and the UI (see page 359)
How to Migrate SSL Keys and Certificates (see page 367)

SSL Administration for the Apache Web Server and the UI

Enable SSL for the following purposes:

- Handling federation traffic across an SSL connection.
- Enabling secure communication across the back channel for HTTP-Artifact single sign-on.
- Enabling secured access to the Administrative UI.

The embedded Apache web server lets the federation system handle SSL federation traffic and secure the back channel for HTTP-Artifact single sign-on. The embedded Tomcat web server allows secured access to the UI.

To enable SSL for the Apache and Tomcat web servers, complete the following process:

1. Create a certificate request for a server certificate.
2. Import the certificate that is issued by the Certificate Authority (CA).
3. Activate SSL in the Administrative UI. Locate the settings in Infrastructure, SSL Configuration.

Note: CA SiteMinder® Federation Standalone installations operating in FIPS Migrate or FIPS-only modes have FIPS-compatible encryption key algorithms available for certificates.

How to Enable SSL for the Apache Web Server and the UI

The procedure for enabling SSL for the embedded Apache web server and the Administrative UI is the same.

- Enable SSL for the embedded Apache web server if you want to do the following:
  - Manage federation traffic across an SSL connection
  - Secure communication across the back channel for artifact single sign-on.

  Remember that an SSL port number is specified when you run the Configuration wizard.
- Enable SSL for the Administrative UI to secure the connection to the UI.

  By enabling SSL, CA SiteMinder® Federation Standalone generates a FIPS-compatible private key for the server certificate.

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://.

To enable SSL communication:

1. Request a server certificate.
2. Specify the CA certificate that signs the server certificate.
3. Upload the signed certificate to the system.

   After the certificate is successfully uploaded, the CA SiteMinder® Federation Standalone activates the SSL connection.

In addition to these required steps, you can do the following:

- Retrieve a certificate signing request.
- Disable SSL.
- Delete the SSL configuration from the system.

Request an SSL Server Certificate

The first step in establishing an SSL connection is to complete a server certificate request. You send the completed request to a trusted Certificate Authority (CA), who returns a signed server certificate.

Important! Request an SSL Server certificate.

Follow these steps:

1. From the Administrative UI, select Infrastructure, SSL Configuration.

   The SSL Configuration dialog opens. In the SSL Configuration Status field, the status reads **Server cert not requested.**

2. Click Request to create a certificate request.

3. Complete the fields in the Certificate Request dialog and click Save.

   Certain fields have required values that are already assigned to them. In the Requester Name field, there is a suggested default value, but you can change it. The Requester Name value must be the fully qualified domain name that is associated with the server where CA SiteMinder® Federation Standalone is deployed.

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

When the certificate request is created, CA SiteMinder® Federation Standalone generates a private key. The private key is stored in an internal file location.

After the request is generated, send the server certificate request to the designated CA that signs the certificate.

Based on the generated certificate request, the Certificate Authority issues a certificate. The validity duration of the certificate validity duration is equal to one of the following values:

- Certificate Authority default value.
- Value that is based on the business agreement between the requester and the Certificate Authority.

Upload the Signed Server Certificate

After you complete a certificate request, the SSL Configuration Status field reads **Server cert requested, not signed**, indicating that the certificate request is waiting to be signed. CA SiteMinder® Federation Standalone accepts a base-64 encoded PEM certificate or a full PKCS #7 certificate/chain response.

After you receive the signed certificate from the CA, the certificate must be uploaded to the storage location.

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

To upload the signed server certificate

1. Begin at the same SSL Configuration where you started the request.
2. Select the signed certificate response in the Signed Certificate Response field. Click Browse to locate the file.

   Note: Only one key and certificate pair is needed for the SSL features because SSL does not support more than one pair.
3. Identify the CA that signed the SSL certificate from the pull-down menu in the CA Certificate field.

   If the CA certificate is not in the key store, import a copy of the CA certificate used to sign the SSL certificate request. Import the certificate by clicking Import and completing the import steps.
4. Click Apply to upload the server certificate to CA SiteMinder® Federation Standalone.

   A confirmation message is displayed and the SSL Configuration changes to reflect that the certificate is now updated.
5. Restart the federation services according to your operating environment.

   - **Windows**

     Use the stop and start shortcuts as follows. If you logged in as a network user and not a local administrator, right-click the shortcut and select Run as administrator.

     a. Start, All Programs, CA, Federation Standalone, Stop services

     b. Start, All Programs, CA, Federation Standalone, Start services

- **UNIX**

  a. Open a command window.

  b. Run the following scripts:

  *federation_install_dir/*fedmanager.sh stop

  federation_install_dir/fedmanager.sh startssl

  Note: Do not stop and start the services as the root user.

After the server certificate is uploaded to the system, CA SiteMinder® Federation Standalone updates the certificate and activates SSL. Assuming that the certificate upload was successful, the SSL Configuration Status reads **SSL Active**. The button in the configuration group box changes to Deactivate.

The UI also indicates whether the uploaded certificate is FIPS-approved or not.

Deactivate SSL

You can deactivate the SSL configuration if you no longer require SSL. For example, if back channel authentication is no longer required or you no longer want an SSL connection to the UI you can deactivate SSL.

Note: If you reconfigure a Windows system with SSL enabled, deactivate the SSL configuration before reconfiguring your system. Reactivate SSL after the reconfiguration is complete.

Follow these steps:

1. Begin at the SSL Configuration dialog.
2. Click Deactivate in the Embedded web server or Administrative UI section.

   A confirmation prompt is displayed asking if you want to disable SSL.
3. Click Yes to complete the deactivation.
4. For the Administrative UI only, delete the tomcat.keystore file manually. This file is located in the following directory:

   federation_install_dir/secure-proxy/SSL/keys

   Deactivating SSL for the Administrative UI does not delete the corresponding key store file. If you change the UI SSL certificate for any reason, the certificate is not updated, which results in CA SiteMinder® Federation Standalone using the wrong certificate. Deleting the Tomcat key store helps ensure that any updates you make to the SSL certificate are reflected.

5. Restart the federation services according to your operating environment.

   - **Windows**

     Use the stop and start shortcuts as follows. If you logged in as a network user and not a local administrator, right-click the shortcut and select Run as administrator.

     a. Start, All Programs, CA, Federation Standalone, Stop services

     b. Start, All Programs, CA, Federation Standalone, Start services

   - **UNIX**

     a. Open a command window.

     b. Run the following scripts:

     federation_install_dir/fedmanager.sh stop

     federation_install_dir/fedmanager.sh start

     Note: Do not stop and start the services as the root user.

The SSL connection is no longer active and the SSL Configuration Status setting changes to **Server cert signed by CA, SSL ready**. The certificate and key files remain so you can re-enable SSL.

Reactivate SSL

If you deactivate SSL for any reason, reactivate it. By enabling SSL, CA SiteMinder® Federation Standalone generates a FIPS-compatible private key for the server certificate.

Note: If the Status setting reads **Server cert signed by CA, SSL ready**, activate the SSL connection.

Follow these steps:

1. Begin at the SSL Configuration dialog.
2. Click Activate in the Embedded web server SSL configuration group box.

   The SSL Configuration Status setting changes to **SSL Active** and a confirmation message is displayed in the dialog.
3. Restart the federation services according to your operating environment.

   - **Windows**

     Use the stop and start shortcuts as follows. If you logged in as a network user and not a local administrator, right-click the shortcut and select Run as administrator.

     a. Start, All Programs, CA, Federation Standalone, Stop services

     b. Start, All Programs, CA, Federation Standalone, Start services

   - **UNIX**

     a. Open a command window.

     b. Run the following scripts:

     *federation_install_dir/*fedmanager.sh stop

     federation_install_dir/fedmanager.sh startssl

     Note: Do not stop and start the services as the root user.

SSL is now enabled. You do not have to modify the SSL configuration until the certificate expires.

Replace or Resubmit a Certificate Signing Request for SSL

You can retrieve a copy of a certificate signing request associated with the private key/certificate pair in use by the Apache server or the UI. The ability to get a copy of the certificate signing request is useful if you delete the request file or you do not save the file. A copy of the request is also useful for resubmitting the request at a later date before the signed certificate expires.

The Retrieve function lets you obtain a copy of the certificate signing request.

Note: The Retrieve option is only available if a certificate has been requested and you have not deleted the SSL configuration by using the Restart button.

To retrieve a certificate signing request

1. From the UI, select Infrastructure, SSL Configuration.

   The SSL Configuration dialog opens.

2. Click Retrieve.

   A File Download dialog opens, prompting you to open or save the file.

3. Save the file.

The signing request is now retrieved and the UI returns to the SSL Configuration dialog.

Remove SSL from the Embedded Apache Server and the UI

- Remove SSL from the embedded Apache web server if you:
  - no longer require a back channel connection for artifact single sign-on
  - no longer want to use SSL
- Remove SSL from the UI connection if you no longer want an SSL connection to the UI.

The Restart feature lets you disable the existing SSL configuration and delete all files associated with the SSL configuration. Specifically, it deletes the private key and server certificate and the original server request file.

To disable SSL and remove related files

1. Log in to the Administrative UI.
2. Select Infrastructure, SSL Configuration.

   The SSL Configuration dialog opens.

3. Click Restart in the group box for the feature that does not require SSL.

   A prompt to confirm the restart is displayed.

4. Click Yes.

   The SSL configuration is removed from the system.

5. Restart the federation services according to your operating environment.

   - **Windows**

     Use the stop and start shortcuts as follows. If you logged in as a network user and not a local administrator, right-click the shortcut and select Run as administrator.

     a. Start, All Programs, CA, Federation Standalone, Stop services

     b. Start, All Programs, CA, Federation Standalone, Start services

   - **UNIX**

     a. Open a command window.

     b. Run the following scripts:

     federation_install_dir/fedmanager.sh stop

     federation_install_dir/fedmanager.sh start

     Note: Do not stop and start the services as the root user.

   Restarting the services lets the Apache web server and the UI return to non-SSL operation. Subsequent HTTPS requests that come in to CA SiteMinder® Federation Standalone will fail. With SSL removed, all service URLs now have to start with http.

How to Migrate SSL Keys and Certificates

For CA SiteMinder® Federation Standalone r12.52, the SSL key and certificate files for the embedded Apache and Tomcat servers are encrypted. For releases 12.0 and 12.0 SP1, these files are not encrypted. To avoid purchasing a new key/certificate pair for an encrypted file, migrate existing key or certificate files from r12.0/r12.0 SP1 to r12.52. You can also export these files for backup purposes without migrating them.

Important! For federation systems before r12.1, the embedded Tomcat server uses a self-signed certificate. You cannot use this self-signed certificate for a migration to r12.52. Purchase a signed certificate and upgrade the Tomcat SSL configuration with the signed certificate.

For Apache, you can migrate files for the SSL connections beginning at r12.0. For Tomcat, you can migrate files only from r12.1 forward because at release12.0, a self-signed certificate secured the Tomcat key store. Beginning with r12.1, the federation system requires that a Certificate Authority signs the certificate.

Migrating SSL keys and certificate files is useful in the following situations:

- To move to a different version of CA SiteMinder® Federation Standalone on a new system instead of upgrading an existing system. Migrate the SSL keys or certificates from the existing system to the new system.
- To migrate SSL keys and certificates from one system in a cluster to another. Migrating lets you reuse the keys and certificates. For example, if a load balancer passes SSL requests to the federation systems in a cluster, each system must use the same keys and certificates. Therefore, you would migrate keys and certificates from one system to the other.

Note: If you upgrade a federation 12.0 system to r12.52, the installer automatically upgrades Apache and Tomcat SSL key and certificate files to encrypted files. This automatic does not apply to migrations.

The certificate and private key files are as follows:

Apache

- The server.key file contains a private key.
- The server.cert file contains a server certificate.

Tomcat

- For r12.0, the tomcat.keystore file contains a self-signed certificate. For r12.1x, the tomcat.keystore file contains a CA-signed certificate and private key pair.

To migrate or export these files, use the CA SiteMinder® Federation Standalone SSL utility named migratessl. The migration utility is included with the product as a batch file for Windows systems and a shell script for UNIX systems. The utility is installed in the federation_install_dir/bin folder.

The process to migrate SSL files is as follows:

1. Copy the key and certificate files from the existing r12 federation system to any location on the r12.52 federation system.
2. Copy the migratessl tool to the location where you copied the key and certificate files.
3. If you migrate signed certificates, export the Certificate Authority certificate that signed the SSL certificate. Before you continue with the migration, import the CA certificate.

Copy Key and Certificate Files from the r12 System

To use the SSL migration tool, first gather the key and certificate files for the CA SiteMinder® Federation Standalone system from which you plan to migrate or export then copy them.

To copy the SSL key and certificate files

1. Locate the files on the existing CA SiteMinder® Federation Standalone system.

   The Apache SSL key and certificate files are in the following locations:

   - *federation_install_dir*/secure-proxy/SSL/keys/server.key
   - *federation_install_dir*/secure-proxy/SSL/certs/server.crt

   The Tomcat SSL key store file is in the following location:

   - *federation_install_dir*/secure-proxy/SSL/keys/tomcat.keystore

2. Copy the key and certificate files to any location on the new CA SiteMinder® Federation Standalone machine.

Copy the SSL Migration Tool to Same Folder as the Key/Certificate Files

The SSL migration tool requires software that is deployed with CA SiteMinder® Federation Standalone 12.1 SP3. Run the tool on the machine where the CA SiteMinder® Federation Standalone 12.1 SP3 product has been installed. Specifically, the tool has to reside in the same folder where you copied the files to be migrated.

To copy the SSL utility tool

1. Navigate to *federation_install_dir*/bin on the r12.52 system.
2. Copy the migratessl file (.bat or .sh) to the location on the r12.52 system where you copied the key and certificate files.

Migrate or Export SSL Keys and Certificates

Complete the SSL key or certificate file migration by running the migratessl utility.

Follow these steps:

1. Import the Certificate Authority certificate that originally signed the SSL certificate you are migrating.

   a. On the system from which you are migrating, export the CA certificate using the Administrative UI.

   b. On the new system to which you are migrating, import the CA certificate using the Administrative UI.

2. Open a command window on the new system where you copied the existing key or certificate files.
3. Navigate to the folder where you copied the components.
4. Specify the migratessl command with the necessary command arguments. Refer to the list of migration tool command arguments for all the options.

   Examples

   - To migrate the SSL server.key for Apache SSL connections, enter:

   ```
   migratessl.bat -op migrate -keytype Apache
   -sourcefile server.key -certfile server.crt
   -sourcever 12.0 -sourceos Windows -oldpwd admin1
   -newpwd admin2 -issueralias trustedca
   ```

   - To migrate a key/cert file for Tomcat SSL connections, enter:

   ```
   migratessl.sh -op migrate -keytype Tomcat
   -sourcefile tomcat.keystore -sourcever 12.1
   -sourceos UNIX -issueralias trustedca
   -oldpwd admin1 -newpwd admin2
   ```

   - To export a key/cert file for Tomcat SSL connections, enter:

   ```
   migratessl.sh -op export -keytype Tomcat
   -sourcefile tomcat.keystore -sourcever 12.1
   -sourceos UNIX  -dest ca/federationmgr/secure-proxy/
   SSL/keys/ -oldpwd admin1 -newpwd admin2
   ```

If you are migrating SSL keys and certificates as part of an entire configuration migration, complete the migration process by reactivating partnerships.

SSL Migration Tool Command Arguments

The migratessl tool is invoked at the command line. When entering a command:

- Follow each command agrument (except for Help flags) by only one value.
- Enclose values that have spaces, such as directory paths in double quotes.

Command Argument	Meaning
-op	Migrate or Export Default: Migrate When exporting for Apache, the tool exports a server.key file and a server.crt file, if you specify the -certfile argument. For Tomcat, the tool exports a tomcat.p12 file, which is a PKCS#12 key/cert file.

-keytype	Apache or Tomcat Default: Apache
-sourcefile	Name of the file containing the SSL key (Apache) or the key store containing the key and certificate (Tomcat).
-certfile	Name of the file containing the Apache SSL server certificate (Apache only).
-sourcever	CA SiteMinder® Federation Standalone version the key or certificate comes from, such as 12.0, 12.1. Default: 12.0
-sourceos	Operating system of the environment the key comes from, Windows or UNIX. **Note:** There is no Linux option because Linux support was introduced in r12.1 SP3. Default: The OS of the machine where the tool is being run.
-dest	Path to the folder for output files. This option is ignored for migration. Default for Export: Current folder **Important!** If you do not specify a destination folder, the files that you are migrating are overwritten.
-issueralias	The alias of the CA certificate that signed the certificate you are migrating. Import the CA certificate under this alias to the destination CA SiteMinder® Federation Standalone system. (Used only for Migrate; ignored for Export.)
-oldpwd	The CA SiteMinder® Federation Standalone administrative password of the system that is the source of the key.
-newpwd	The CA SiteMinder® Federation Standalone administrative password of the system to which the key is being moved.
-h	Displays these usage instructions.
-help	Displays these usage instructions.
-?	Displays these usage instructions.

Chapter 28: Logs to Monitor Federation Activities

This section contains the following topics:

Federation Logging Overview (see page 373)
Federation Web Services (FWS) Logging (see page 374)
Server Trace Logging (see page 376)
server.log File Setup (see page 379)
Federation Data Object Trace Logging (see page 384)
Audit Logging (see page 385)
Transaction IDs to Aid Federation Troubleshooting (see page 392)

Federation Logging Overview

Troubleshoot federation operation by enabling logging. Logs provide important diagnostic information for you and for CA Support.

Several logs provide information about federated activities. By default, the system enables the following logs:

- Federation Web Services (FWS) application logs, which include:

  affwebservices.log—This log file contains messages about the Federation Web Services (FWS) application. The default path for this file is *federation_install_dir*\logs\fws.

  FWSTrace.log—This trace log contains information about FWS runtime activities.

- Server logs for the policy server that is used by the federation product, which include:

  smtracedefault.log—This trace log follows server runtime activities. The default location for this trace log is in the directory *federation_install_dir*\logs\server.

  Note: If you enable tracing, large log files can result.

  smps.log—This log file contains informational and trace messages about the server. This log file is in the directory *federation_install_dir*\logs\server.

- Administrative UI operational log

  server.log—This log file contains messages about the Administrative UI and the embedded SPS server. This log file is in the directory *federation_install_dir*\logs\ui.

You can also enable the federation data store objects trace log, **XPSConfig_***date_time_stamp*.log. This trace log monitors trace activity for the federation objects in the data store.

Checkpoint Log Messages

In the FWSTrace.log and the smtracedefault.log, there are checkpoint log messages that indicate what is happening during a transaction. For example:

```
[07/30/2013][11:34:44][4260][5824][1181adbb-993f775c-33ba08f3-76b52f3b-3d2280cd-4ae][SSO.java][processRequest][Reading SAML 2.0 SP Configuration [CHECKPOINT = SSOSAML2_SPCONFREAD_REQ]
```

You can search on these checkpoint messages to follow some of the processes occurring during a transaction.

In addition to the checkpoint messages, there are transaction IDs (see page 392), which you can use to follow a transaction. If a transaction fails, the checkpoint messages and transaction IDs can help you determine the specific problem.

Federation Web Services (FWS) Logging

You can monitor the FWS application run-time activities by enabling the following logs:

- Informational logging (affwebserv.log)
- Trace logging (FWSTrace.log)

To manage the log behavior, such as rollover frequency and log size, modify settings in the LoggerConfig.properties file.

Note: Do not confuse the LoggerConfig.properties file with the logger.properties file that configures the server.log file. The names are similar, but the files are different.

Follow these steps:

1. Navigate to the following directory:
   federation_install_dir\secure-proxy\Tomcat\webapps\affwebservices\WEB-INF\classes\

   Note: Use a forward slash (/) in the paths for a UNIX operating environment.
2. Open the LoggerConfig.properties file in a text editor.
3. (Optional) Modify the log settings. Review explanations and options for each setting in the LoggerConfig.properties file. The settings are:

   LoggingOn

   Enables or disables informational logging.

   LogFileName

   Default: *federation_install_dir*\\logs\\fws\\affwebserv.log

   affwebserv.log is the default file name. You can change the name.

LogLocalTime

LogRollover

LogSize

LogCount

4. (Optional) Modify the trace settings for logging FWS messages. Review the explanations and the options for each setting in the LoggerConfig.properties file.

   TracingOn

   Enables or disables FWS trace logging to the FWSTrace.log file.

   EnableDNSLookUp

   TraceFileName

   The default output file name is FWSTrace.log. You can change this name.

   TraceConfigFile

   Identifies the trace configuration file. The configuration file determines which components and subcomponents the system monitors and logs messages.

   TraceRollover

   TraceSize

   TraceCount

   TraceFormat

   TraceDelim

5. Save and close the file.
6. Restart the federation services according to your operating environment.

   - **Windows**

     Use the stop and start shortcuts as follows. If you logged in as a network user and not a local administrator, right-click the shortcut and select Run as administrator.

     a. Start, All Programs, CA, Federation Standalone, Stop services

     b. Start, All Programs, CA, Federation Standalone, Start services

   - **UNIX**

     a. Open a command window.

     b. Run the following scripts:

     federation_install_dir/fedmanager.sh stop

     federation_install_dir/fedmanager.sh start

     Note: Do not stop and start the services as the root user.

Server Trace Logging

The server trace log (smtracedefault.log file) follows federation server runtime activities. The default location for this trace log is in the directory *federation_install_dir*\logs\server.

Note: If you enable tracing, large log files can result.

Two tasks are required to set up server-side trace logging:

1. Set up the server trace log configuration file. (see page 376) The configuration file defines which components are monitored and written to the smtracedefault.log file. You can use the default file, smtracedefault.txt or use one of the other templates provided.
2. Configure the behavior of the server trace log file, smtracedefault.log (see page 377). Specify the location of the log output file, the location of the log configuration file, the format of the log output file, and the log rollover frequency.

Set Up the Server Trace Log Configuration File

Set up the log configuration file. A log configuration file defines which components are monitored and therefore, what is written to the smtracedefault.log file. You can use one of the following files for federation:

- smtracedefault.txt (default)
- samlidp_trace.template (activity at the asserting party)
- samlsp_trace.template (activity at the relying party)

For efficiency, use one of the templates. Enter the name of a template in the LogTraceConfig parameter, which you access with the XPSConfig command and by selecting the SM option.

Note: The preconfigured templates are in *federation_install_dir*\siteminder\config\profiler_templates.

In place of the templates, you can use the default file and manually add all federation components in this file.

Follow these steps:

1. Navigate to *federation_install_dir*\siteminder\config\smtracedefault.txt.
2. Back up the template file.

3. Open the smtracedefault.txt file in an editor.
4. Edit the file by copying and pasting the following text into the file. Overwrite the existing text.

```
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/Assertion_Generator, Fed_Server/Auth_Scheme,
Fed_Server/Configuration
data: Date, Time, Tid, TransactionID, SrcFile, Function, Domain,
Resource, Action, User, SessionID, Data, AuthReason, Message
```

5. Save the file.

Configure the Behavior of the Server Trace Log File

Use the XPSConfig tool, included with the product, to enable tracing for server-side runtime activities. XPSConfig is an interactive command-line utility that allows you to view product parameters and edit their settings.

Follow these steps:

1. Open a command window on the system where the product is installed.
2. Enter XPSConfig.

   Type the command as it is shown here. The command is case-sensitive.

   The Products Menu displays.
3. Enter SM.

   The Parameters Menu displays. A number is associated with each parameter that you can modify.
4. Enter the number associated with the parameter you want to modify. LogTraceConfig parameter.

5. Enter c to change the value.
6. Enter q to return to the parameter list.
7. For server tracing, specify values for the following settings.

LogTrace

Enables tracing. The default for tracing is to be off, which is indicated by a space between two quotation marks (" "). Do not leave the setting blank.

LogTraceConfig

This value points to the tracing configuration file. There is no default value. Enter *federation_install_dir*\siteminder\config*template_name.* The default template is smtracedefault.txt. There are other trace templates specifically for federation.

Examples:

federation_install_dir\siteminder\config\samlidp_trace.template

federation_install_dir\siteminder\config\samlsp_trace.template

LogTraceConsole

Indicates whether messages are displayed in a console window. The default is not to log to the console.

LogTraceFormat

Determines how information is displayed in the log. The default is sm. Works with the LogTraceDelimiter setting to specify which character acts as a delimiter.

LogTraceMode

Specifies the trace mode. The default is 0.

LogTraceDelimiter

Identifies a character to serve as a delimiter in the log output file. There is no default value.

LogTraceOutput

Specifies the location of the log output file. The default is *federation_install_dir*\logs\server.

8. Configure how frequently the logging and tracing files rollover by modifying the rollover settings. Enter the number associated with the parameter you want to modify.

   Note: Any changes to the log rollover settings apply to the smtracedefault.log file and the smps.log file.

   The rollover parameters are as follows:

   LogFilesToKeep

   Represents the number of Policy Server error logs to keep. Older files are deleted older files.

   LogRolloverDays

   Indicates if a rollover occurs on a daily basis. Enter the number that corresponds to the number of days that pass before a rollover occurs.

   LogRolloverInterval

   Indicates if a rollover occurs on an hourly basis. If this value is set, LogRolloverDays is ignored.

   LogRolloverOnStart (enabled by default)

   Indicates whether the log file is rolled over when the services are started.

   LogRolloverSize

   Indicates at what size the log files is rolled over. If the system reaches the size limit before the next rollover interval, the log file still rolls over.

   LogRolloverTime

   Indicates what time of day to perform the rollover. The system uses this setting with the LogRolloverDays parameter. Enter a value in the form "hour:minutes" using a 24 hour clock.

   Example: "22:00"

9. If you are finished configuring parameters, keep entering q until you exit XPSConfig.

Changes made in XPSConfig are not recognized until you exit the XPSConfig tool. Where noted, some changes require that you restart the system services.

server.log File Setup

The server.log file is useful for examining the Administrative UI operation of the product. This log also contains messages about the embedded SPS server. This log file resides in the directory *federation_install_home*/logs/ui.

The **logger.properties** file and the **log4j.properties** file contain the log settings that determine what is recorded in the server.log file. The settings are groups of name/value pairs or directives that the system reads at run time.

Logger.properties file

The logger.properties file resides in the directory *federation_install_dir*/secure-proxy/Tomcat/properties. The file contents are grouped into the following sections:

- SvrConsoleAppender settings
- SvrFileAppender settings
- Server.conf settings
- Log rollover settings

The directives contained in this file follow the format name=value. Any lines beginning with the # symbol are comments, and are not read when the system loads configuration settings.

Note: Pathnames on Windows systems use double backslashes (\\).

Log4j.properties

The lo4j.properties file resides in the directory *federation_install_dir*/secure-proxy/Tomcat/webapps/fedui/WEB-INF/classes. This file determines the log level recorded for Administrative UI operation.

The procedure for modifying the log files is the same. You can modify the files without restarting the system.

Follow these steps:

1. Open the file in a text editor.
2. Edit the directives, as necessary.
3. Save the file.

The log settings are changed.

Log Settings

Server.conf Settings

The Server.conf settings in the logger.properties file enable and disable logging, sets logging level, and sets the output format of the log messages. The entry that you can modify in this section must be in the following format:
`log4j.rootCategory=<log_level>,<output_format>`

log_level

Specifies the log level of a message. The following values are listed in an increasing order of priority:

OFF, FATAL, ERROR, WARN, INFO, DEBUG, ALL

To disable logging, set the log level to OFF. If the value is set to any other value, logging is enabled.

Default: INFO

output_format

Specifies whether a log message is displayed in a console, or in a file, or both.

Default: SvrFileAppender

Example: To set the log level to INFO and display a log message in a console and in a file, use the following entry:
`log4j.rootCategory=INFO,SvrConsoleAppender,SvrFileAppender`

SvrConsoleAppender Settings

The SvrConsoleAppender Settings section controls logging events to a console. The entry that you can modify in this section is:
`log4j.appender.SvrConsoleAppender.layout.ConversionPattern=<log_message_format>`

log_message_format

Specifies the format of a log message output to the console. The product supports all the log4j date pattern strings.

Default Value: [%d{dd/MMM/yyyy:HH:mm:ss-SSS}] [%p] - %m%n

SvrFileAppender Settings

The SvrFileAppender Settings section controls logging events to a file. This section defines the log rollover frequency and the format of the log messages that are written to the file. The entries that you can modify in this section are:

```
log4j.appender.SvrFileAppender.File=<log_file_path>
log4j.appender.SvrFileAppender.Append=true
log4j.appender.SvrFileAppender.layout.ConversionPattern=<log_message_format>
```

log_file_path

Specifies the name and path of the log file.

Default Name: server.log

Default Path: *install_dir_home*/secure-proxy/proxy-engine/logs/ui/server.log

true|false

Directs the system whether to append log messages to an existing file. If this value is set to true, the system appends new log messages to the existing log file. If this value is set to false, the system rolls over the existing log file and generates a new log file.

Default Value: true

log_message_format

Specifies the format of log messages that the system writes to the server.log file. The product supports all the log4j date pattern strings.

Default Value: [%d{dd/MMM/yyyy:HH:mm:ss-SSS}] [%p] - %m%n

Type of Log Rolling Used

The log rolling section determines when an existing log file is rolled over and a new log is generated. Enable a log to rollover based on the file size or the file date.

The entries that you can modify in this section are:

```
log4j.appender.SvrFileAppender.MaxFileSize=1MB
log4j.appender.SvrFileAppender.MaxBackupIndex=10
#log4j.appender.SvrFileAppender.DatePattern='.'yyyy-MM-dd
```

MaxFileSize

Specifies the maximum size the log file can reach after which the system must create a log file.

Default Value: 1MB

MaxBackupIndex

Specifies the maximum number of log files that the system creates. If the number of log files exceed the MaxBackupIndex number, the system deletes the oldest log file and it generates a new one.

Default Value: 10

DatePattern

Specifies the date when the system must create a log file.

Default: yyyy-MM-dd

A new log file is created with the name <*log_file_name*>.<*date_format*>

log_file_name

Specifies the name of the log file.

Default: server.log

date_format

Specifies the date when the log file was created. The file supports all the log4j date pattern strings.

Default: yyyy-MM-dd

log4j.properties File for server.log

The log4j.properties file controls extra Administrative UI logging that the system writes to the server.log file. This file resides in the directory *federation_install_dir*\secure-proxy\Tomcat\webapps\fedui\WEB-INF\classes.

You can modify the following entries:
`log4j.appender.UIConsoleAppender.layout.ConversionPattern=<log_message_format>`

log_message_format

Specifies the format of the log message output to the console. The product supports all the log4j date pattern strings.

Default Value: [%p] %c - %m%n

`log4j.rootCategory=<log_level>,<output_format>`

log_level

Specifies the log level of a message. The following values are listed in an increasing order of priority:

OFF, FATAL, ERROR, WARN, INFO, DEBUG, ALL

To disable logging, set the log level to OFF. If the value is set to any other value, logging is enabled.

Default: INFO

output_format

Specifies whether a log message is output to a console, or in to file, or both.

Default: UIConsoleAppender

Example: To set the log level to INFO and display a log message in a console and in a file, use the following entry:
`log4j.rootCategory=INFO,UIConsoleAppender,UIFileAppender`

You can also uncomment the two DEBUG entries.

Federation Data Object Trace Logging

Enable XPS tracing to monitor federation data store objects. These activities are written to the smtracedefault.log. The smtracedefault.log is in the directory *federation_install_dir*\logs\server.

Follow these steps:

1. Open a command window.
2. Enter XPSConfig.

   Type the command as it is shown here. The command is case-sensitive.

   The Products Menu displays.
3. Enter X for the xTrace option.

   The Tracer Menu displays.
4. Enter the number that is associated with the **fed** option. All options that are related to fed are selected and marked with an "x."
5. Enter U to save the selections. This updates the tracer menu.
6. Enter q until you exit the XPSConfig tool.
7. Restart the federation services so that the changes take effect.

8. Restart the federation services according to your operating environment.

   - **Windows**

     Use the stop and start shortcuts as follows. If you logged in as a network user and not a local administrator, right-click the shortcut and select Run as administrator.

     a. Start, All Programs, CA, Federation Standalone, Stop services

     b. Start, All Programs, CA, Federation Standalone, Start services

   - **UNIX**

     a. Open a command window.

     b. Run the following scripts:

     federation_install_dir/fedmanager.sh stop

     federation_install_dir/fedmanager.sh start

   Note: Do not stop and start the services as the root user.

Audit Logging

CA SiteMinder® Federation Standalone automatically creates an audit log, smaccess.log, located in the directory *federation_install_dir*/logs/server. This log remains empty until you enable logging for authentication events or authorization events, or both, using the XPSConfig command.

Note: XPSConfig is case-sensitive on UNIX platforms.

To enable audit logging

1. Open a command window.
2. Type XPSConfig at the command prompt.

   The Product Menu is displayed.
3. Enter SM.

   The list of parameters with their current values is displayed.
4. (Optional) Enter **f** to filter the list of settings.

   At the Enter Filter prompt enter **report** to find all the settings related to audit log.
5. Enter the number associated with the type of audit logging to enable.

   ReportAuth

   Specifies the log settings for authentication events.

   ReportAz

   Specifies the log settings for authorization events.

6. Enter c to change the value. The default is 0, which means that no events are logged.
7. Enter one of the following values at the prompt:

   1 = log all events

   2 = log only rejection events
8. Enter q until you return to the Products Menu.

   Audit logging is enabled.

Note: You can repeat this procedure at any time to update the settings for the audit log settings.

Set the Audit Log Name and Location (Optional)

The default name for the audit log is smaccess.log, and the default location is *federation_install_dir*/logs/server. You can change these values.

Follow these steps:

1. Open a command window.
2. Type XPSConfig at the command-line prompt.

   Note: XPSConfig is case-sensitive on UNIX platforms.

   The Product Menu is displayed.
3. Enter SM.

   The list of parameters and their values is displayed.
4. (Optional) Enter **f** to filter the list of settings.

   At the Enter Filter prompt enter **text** to find the setting related to audit log text file name.
5. Enter the number associated with the ReportTextFile setting.

   The current value is displayed.
6. Enter c to change the file name.
7. Enter the a valid path and a new file name.
8. Enter q until you return to the system command prompt.

The new file name and location is saved.

Use an ODBC Database for Audit Logging (Optional)

You can use an ODBC database to record audit data instead of the default text file.

Follow these steps:

1. Change the audit log storage type to ODBC.
2. Configure an ODBC data source. Refer to one of the following set of instructions:
   - Create a SQL Server Data Source on Windows (see page 388)
   - Create a SQL Server Data Source on UNIX Systems (see page 389)
   - Create an Oracle Data Source on Windows (see page 390)
   - Create an Oracle Data Source on a UNIX System (see page 391)

Change the Audit Log Storage Type

The audit log is in text format, by default. To store audit data in an ODBC database, change the storage type of the log.

Important! If you change the audit log storage type from TEXT to ODBC, you cannot change it back.

Follow these steps:

1. Open a command window.
2. Type XPSConfig at the command-line prompt.

   Note: XPSConfig is case-sensitive on UNIX platforms.

   The Product Menu is displayed.
3. Enter SM.

   The list of parameters and their current values is displayed.
4. (Optional) Enter **f** to filter the list of settings.

   At the Enter Filter prompt enter **store** to find all the settings that are related to the audit log storage type.
5. Enter the number for the LogStoreNamespace setting.

   The current value is displayed.
6. Enter c to change the storage type.
7. Enter **ODBC:** at the prompt.

   Note: Include the colon in the entry.
8. Type q twice to return to the list of parameters.

9. Configure these additional settings. Enter the number for each setting to modify it.

   Note: Enter **f** to filter the list of settings. At the Enter Filter prompt, enter **Db** to find all the settings related to audit log database.

   DbLogAdminName

   Specifies the data source user name for the audit log.

   Limits: A string; only applies when LogStoreNamespace is set to ODBC:.

   DbLogAdminPassword

   Specifies the data source user password for the audit log.

   Limits: A string; only applies when LogStoreNamespace is set to ODBC:.

   DbLogDataSource

   Specifies the data source name for the audit log.

   Limits: A string; only applies when LogStoreNamespace is set to ODBC:.

   DbLogMaxConnections

   Specifies the maximum number of connections to the data source for the audit log.

   Default: 15

   Limits: Must be an integer; only applies when LogStoreNamespace is set to ODBC:.

   DbLogUseDefault

   Specifies whether the audit log will use the same ODBC data source as the policy store.

   Default: FALSE

   Limits: TRUE or FALSE; only applies when LogStoreNamespace is set to ODBC:.

10. Enter q enough times to return to the system command prompt.
11. To use an ODBC database to record audit data, set up a data source (see page 387).

Create a SQL Server Data Source on Windows

ODBC requires that you configure a data source for the MS SQL Server wire protocol.

To create the data source on Windows

1. Do one of the following:
   - If you are using a supported 32–bit Windows operating system, click Start and select Programs, Administrative Tools, ODBC Data Sources.

- If you are using a supported 64–bit Windows operating system:
  a. Navigate to the *install_home*\Windows\SysWOW64.
  b. Double–click odbcad32.exe

The ODBC Data Source Administrator appears.

2. Click the System DSN tab.

   System data source settings appear.

3. Click Add.

   The Create New Data Source dialog appears.

4. Select CA SiteMinder® SQL Server Wire Protocol and click Finish.

   The ODBC SQL Server Wire Protocol Driver Setup dialog appears.

5. Enter the data source name in the Data Source Name field.

   Example: CA SiteMinder® Federation Standalone Data Source.

   Note: Take note of your data source name. This information is required as you configure your database as a policy store.

6. Enter the name of the MS SQL Server host system in the Server field.

7. Enter the database name in the Database Name field.

8. Click Test.

   The connection settings are tested and a prompt appears specifying that the connection is successful.

9. Click OK.

   The SQL Server data source is configured and appears in the System Data Sources list.

Create a SQL Server Data Sources on UNIX Systems

The CA SiteMinder® Federation Standalone ODBC data sources are configured using a system_odbc.ini file, which you create by renaming sqlserverwire.ini, located in *federation_install_dir*/*siteminder*/db, to system_odbc.ini. This system_odbc.ini file contains all of the names of the available ODBC data sources as well as the attributes that are associated with these data sources. This file must be customized to work for each site. Also, you can add additional data sources to this file, such as defining additional ODBC user directories for CA SiteMinder®.

The first section of the system_odbc.ini file, [ODBC Data Sources], contains a list of all of the currently available data sources. The name before the "=" refers to a subsequent section of the file describing each individual data source. After the "=" is a comment field.

Note: If you modify of the first line of data source entry, which is [CA SiteMinder® Data Source], take note of the change because you will need this value when configure your ODBC database as a policy store.

Each data source has a section in the system_odbc.ini file describing its attributes. The first attribute is the ODBC driver to be loaded when this data source is used by CA SiteMinder®. The remaining attributes are specific to the driver.

Adding a MS SQL Server Data source involves adding a new data source name in the [ODBC Data Sources] section of the file, and adding a section that describes the data source using the same name as the data source. You need to change the system_odbc.ini file if you create a new service name or want to use a different driver. You should have entries for the Oracle or SQL drivers under [CA SiteMinder® Data Source].

Again, to configure a MS SQL Server data source, you must first create a system_odbc.ini file in the *federation_install_dir*/*siteminder*/db directory. To do this, you need to rename sqlserverwire.ini, located in *federation_install_dir*/*siteminder*/db, to system_odbc.ini.

Create an Oracle Data Source on Windows

Create an ODBC data source for an Oracle database.

Follow these steps:

1. Do one of the following:
   - If you are using a supported 32–bit Windows operating system, click Start and select Programs, Administrative Tools, ODBC Data Sources.
   - If you are using a supported 64–bit Windows operating system:
     a. Navigate to the *install_home*\Windows\SysWOW64.
     b. Double–click odbcad32.exe

   The ODBC Data Source Administrator appears.
2. Click the System DSN tab, and then click Add.

   The Create New Data Source dialog appears
3. Select CA SiteMinder® Oracle Wire Protocol, and click Finish.

   The ODBC Oracle Wire Protocol Driver Setup dialog appears. The General tab is pulled to the front.
4. Enter a name that identifies the data source in the Data Source Name field.

   Note: Record this name. You will need the data source name when pointing the Policy Server to the database.

5. Enter the machine name where the Oracle database is installed in the Host Name field.
6. Enter the port number where the Oracle database is listening on the machine in the Port Number field.
7. Enter the name of the Oracle instance to which you want to connect in the SID field.

   Note: The service name is specified in the tnsnames.ora file. The SID is the system identifier for the database instance. The tnsnames.ora file contains service names and details that Oracle uses to identify and connect to Oracle instances.

   Example: if the tnsnames.ora file contains the following entry for an Oracle instance, you enter instance1 in the SID field:

   ```
   instance1 =
       (Description=
       (Address = (PROTOCOL = TCP)(Host = myhost)(Port=1521))
       (Connect_DATA_ = (SID = SIDofinstance1))
       )
   ```

8. Click Test Connection.

   The connection settings are tested and a prompt appears specifying that the connection is successful.

9. Click OK.

   The Oracle data source is configured for the wire protocol driver.

Create an Oracle Data Source on a UNIX System

The CA SiteMinder® ODBC data sources are configured using a system_odbc.ini file, which you create by renaming oraclewire.ini, located in *federation_install_dir*/*siteminder*/db, to system_odbc.ini. This system_odbc.ini file contains all of the names of the available ODBC data sources as well as the attributes that are associated with these data sources. This file must be customized to work for each site. Also, you can add additional data sources to this file, such as defining additional ODBC user directories for CA SiteMinder®.

The first section of the system_odbc.ini file, [ODBC Data Sources], contains a list of all of the currently available data sources. The name before the "=" refers to a subsequent section of the file describing each individual data source. After the "=" is a comment field.

Note: If you modify of the first line of data source entry, which is [CA SiteMinder® Data Source], take note of the change because you will need this value when configure your ODBC database as a policy store.

Each data source has a section in the system_odbc.ini file describing its attributes. The first attribute is the ODBC driver to be loaded when this data source is used by CA SiteMinder®. The remaining attributes are specific to the driver.

Adding an Oracle Data source involves adding a new data source name in the [ODBC Data Sources] section of the file, and adding a section that describes the data source using the same name as the data source. You need to change the system_odbc.ini file if you create a new service name or want to use a different driver. You should have entries for the SQL Server or Oracle drivers under [CA SiteMinder® Data Source].

Again, to configure an Oracle data source, you must first create a system_odbc.ini file in the *federation_install_dir*/*siteminder*/db directory. To do this, you need to rename oraclewire.ini, located in *federation_install_dir*/*siteminder*/db, to system_odbc.ini.

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-bbbede33-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 Standalone 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 Standalone 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.

How To Follow a Single Transaction in a Log

To monitor a transaction, you can follow the two types of transaction IDs in the FWSTrace.log or smtracedefault.log. If there is a failure, looking at the IDs can help you determine the failure point.

To follow a transaction in a log, use one or more of the following methods:

- Open the trace file in a text editor and search on the string **SAMLTransactionID** (no spaces) or search for a sepcific SAMLTransactionID. This collection of entries in the log provides a view of the entire end-to-end transaction. You can see how far a transaction proceeded.

- Follow the transaction ID in the log file. The transaction ID represents HTTP transactions. Multiple transaction IDs can be associated with a single SAML Transaction ID. A failed transaction displays the transaction ID in the browser. To search the FWSTrace.log and smtracedefault log for the checkpoint error messages, use the displayed transaction ID.
- Parse the log files with a tool that searches files. On UNIX and Windows platforms, you can use a tool like the grep command. The grep command can stream through raw data, line by line, without your having to load a large text file into a text editor.

  Example:

  ```
  [usr@rhel632 etc]# more fwstrace.log| grep checkpoint
  [CHECKPOINT = SSOSAML2_SPCONFFROMPS_REQ]]
  [CHECKPOINT = SSOSAML2_SPCONFREAD_REQ]]
  [CHECKPOINT = SSOSAML2_SPCONFFROMCACHE_REQ]]
  [CHECKPOINT = SSOSAML2_SESSIONCOOKIEVALIDATE_REQ]]
  ```

Chapter 29: Restore Federation System Configuration

This section contains the following topics:

How To Restore a System to a Previous Configuration (see page 395)

How To Restore a System to a Previous Configuration

You can restore a system configuration by reverting to a previously backed-up configuration. If a current configuration is experiencing problems, reverting to a previous configuration can be useful.

The process for reverting to a previous configuration is as follows:

1. Back up the existing CA SiteMinder® Federation Standalone configuration for the system that you want to restore.
2. Run the Configuration wizard on this system using the same settings that were in place when you created the backup configuration.

   When you run the Configuration wizard, the settings must remain the same.

   The following settings must match the original configuration:

   - **Deployment Settings**

     Select the same deployment mode (proxy or standalone) for the new system.

   - **Port numbers**

     Specify the same ports that the backed-up system uses.

   - **Virtual Host Name**

     If the system used a virtual host when initially configured, use the same virtual host name. Additionally, make the appropriate entries in the host file for the system.

   - **SiteMinder Connector**

     If the system used the CA SiteMinder® Connector, select the CA SiteMinder® Connector again.
3. Import the backed-up configuration to the system.

The following sections detail the process.

Back up an Existing Configuration

A backup of your configuration is useful for to recover or migrate your federation system.

Note: This procedure applies to version r12.52 and higher.

To back up a configuration, export the configuration data. The XPSExport tool, included with the product, lets you export the configuration data to an XML file.

Important! During the export process, federation transactions cannot process successfully.

Follow these steps:

1. Open a command window.
2. Enter the following command to export the configuration:

   XPSExport *export_file_name* -xe -xp -passphrase *passphrase*

   export_file_name

   Names the output file that results from the export. The output from XPSExport is in XML format, therefore, the file name must end with the extension **.xml**.

   passphrase

   Specifies the passphrase that is required to encrypt sensitive data. The passphrase must be at least eight characters and must contain at least one digit, one uppercase and one lowercase letter. If the passphrase contains a space, then it must be enclosed in quotes.

   NOTE: If you do not want to enter the passphrase directly, leave it off the command. XPSExport then prompts you for a passphrase and a passphrase confirmation, which is not echoed to the screen.

The export produces an XML file that contains encrypted configuration data. Use this file to restore the configuration.

Revert to a Backed-up Configuration

If you experience problems with an existing CA SiteMinder® Federation Standalone configuration, revert to a previously backed up configuration on the same system.

To restore a configuration, use the XPSImport tool shipped with the product to import an XML file.

Important! Follow the import steps exactly as outlined. Do not access the Certs & Keys tab in the Administrative UI until the procedure is complete.

Follow these steps:

1. Establish a new database instance for federation data.

   Important! Do not use an existing database for this step. The import fails if you do.

2. Run the Configuration Wizard, specifying the new database instance when prompted.

   Use the same settings for this new configuration that were used for the original configuration. These settings include:

   - Deployment Mode
   - Port numbers
   - Virtual Host Name
   - CA SiteMinder® Connector

3. Stop the federation services according to your platform.

   Windows

   Use the stop shortcut. If you logged in as a network user and not a local administrator, right-click the shortcut and select Run as administrator.

   Select Start, All Programs, CA, Federation Standalone, Stop services.

   UNIX

   a. Open a command window.

   b. Run the following script:

      federation_install_dir/fedmanager.sh stop

      Note: Do not stop and start the services as the root user.

4. Restore all configuration data using the XPSImport command:

   XPSImport *export_file_name* -passphrase *passphrase*

   export_file_name

   Names the XML file that resulted from the export of the original configuration. The filename should end with the extension **.xml**.

   passphrase

   Specifies the passphrase required to decrypt sensitive data. It must be at least eight characters and must contain at least one digit, one upper case and one lower case letter. If the passphrase contains a space, then it must be enclosed in quotes.

5. Re-run the Configuration Wizard.

   Use the same settings for this new configuration that were used for the original configuration. These settings include:

   - Deployment Mode
   - Port numbers
   - Virtual Host Name
   - SiteMinder Connector

6. (Optional) If the SiteMinder Connector was enabled in the original configuration, reestablish the Connector by doing the following:

   a. Log in to the Administrative UI.

   b. Click the Infrastructure tab and select Deployment Settings.

   c. Reconfigure the Connector settings using the same values that were used by the original configuration.

   d. Click Register Host to reregister the federation system with the Policy Server.

The configuration is restored to its original state.

Chapter 30: Troubleshooting

This section contains the following topics:

System Performance Troubleshooting (see page 399)
Resolving Signature Verification Failures (see page 401)
Two SSO Transactions Fail Using the Same Browser Session (see page 402)
Examine Secure Proxy Engine Logs to Troubleshoot the System (see page 402)

System Performance Troubleshooting

The following issue describes system performance troubleshooting.

Configure the Session Store Timeout for Heavy Load Conditions

Under heavy load conditions, long-running queries necessary for session store maintenance tasks, such as removing idled–out or expired sessions, can timeout. Adjust the timeout for session store maintenance tasks (60 seconds by default), by increasing the value of the MaintenanceQueryTimeout registry setting. Increase the value so that the maintenance thread can complete its tasks successfully.

The MaintenanceQueryTimeout registry setting can be found at the following registry location:

```
HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\Netegrity\SiteMinder\CurrentVersion\
SessionServer
```

Proxy Engine Hangs and Stops Processing Requests

Symptom:

After processing requests for several days, CA SiteMinder® Federation Standalone's built-in proxy engine hangs.

Solution:

Modify tuning parameters in the proxy engine's server.conf for the connection between Apache web server (HTTP listener) and the proxy engine (Tomcat servlet engine).

The parameters you modify are used by the component mod_jk, which acts as the Tomcat connector to enable communication between the Apache web server and Tomcat using the Apache JServ protocol (AJP).

To modify the server.conf file

1. Navigate to the following directory:

   federation_install_dir/secure-proxy/proxy-engine/conf

2. Open the server.conf file in an editor.

3. Modify the following parameters

   worker.ajp13.reply_timeout

   Specifies the maximum time, in milliseconds, that can elapse between any two packets received from the proxy engine. After this timeout expires, the connection between the Apache server (HTTP listener) and the proxy engine is dropped. A value of 0 indicates that the proxy engine will wait indefinitely until a response is received.

   To ensure that the connection does not wait indefinitely for a response from the proxy engine, increase this value.

   Default: 0

   worker.ajp13.retries

   Indicates the maximum number of times that the mod_jk component sends a connection request to the proxy engine in case of a communication error. After the number of retries has been met and there is no response from the proxy engine, the connection is dropped.

   Increase this value for more retry attempts for a connection request.

   Default: 2

4. Save the server.conf file.

Resolving Signature Verification Failures

A malicious user can commit an XML signature wrapping attack by changing the content of a document without invalidating the signature. By default, software controls for the Policy Server and Web Agent Option Pack are set to defend against signature wrapping attacks. However, a third-party product can issue an XML document in a way that does not conform to XML specifications. As a result, the default signature checks can result in a signature verification failure.

Signature verification failures occur for the following reasons:

- A duplicate ID element is in the XML document and the signature references this duplicate ID. Duplicate ID attributes are not permitted.
- The XML signature does not reference the expected parent element, and a signature wrapping vulnerability is logged.

If a federation transaction fails, examine the smtracedefault.log file and the fwstrace.log file for a signature verification failure. These errors can indicate that the received XML document is not conforming to XML standards. As a workaround, you can disable the default Policy Server and Web Agent protection against signature wrapping attacks.

Important! If you disable the protection against signature vulnerabilities, determine another way to protect against these attacks.

To disable the XML signature wrapping checks:

1. Navigate to the xsw.properties file. The file exists in different locations for the Policy Server and the Web Agent.
   - For error messages in the Policy Server smtracedefault.log file, go to *siteminder_home*/config/properties
   - For error messages in the Web Agent fwstrace.log, go to

     *web_agent_option_pack_ home/*affwebservices/web-INF/classes.

     Note: If the web agent option pack is installed on the same system as the web agent, the file resides in the *web_agent_home* directory.
2. Change the following xsw.properties settings to true:
   - DisableXSWCheck=true (Policy Server setting only)
   - DisableUniqueIDCheck=true (Policy Server and Web Agent Option Pack setting)

     Note: The value of the DisableUniqueIDCheck setting must be the same for the Policy Server and the Web Agent Option Pack.
3. Save the file.

Two SSO Transactions Fail Using the Same Browser Session

Symptom:

A user attempts two single sign-on transactions in the same browser session. The transactions are from the same asserting party to different relying parties. The first transaction is successful but the second transaction results in an authorization failure at the asserting party. The failure occurs because the two partnerships are configured to use different asserting party user directories.

When CA SiteMinder® Federation Standalone initiates a single sign-on transaction at the asserting party, it places a session cookie in the browser. This session cookie contains information about the user ID and the asserting party user directory. Only one CA SiteMinder® Federation Standalone session cookie can exist in the browser at a time.

When a user attempts a second transaction in the same browser session as the first transaction, the session cookie for the first transaction remains in the browser. However, this session cookie does not have the correct information for the second partnership and the authorization operation fails.

Solution:

Use the same browser session for different single sign-on transactions only if the asserting party user directory for each partnership is the same.

If different asserting party user directories are configured for each partnership, close the first browser session and start a new browser session to attempt the second transaction.

Examine Secure Proxy Engine Logs to Troubleshoot the System

Partnership-based CA SiteMinder® Federation Standalone contains a secure proxy engine that forwards traffic to backend servers. The secure proxy engine includes the following components:

- Apache Web Server

  Acts as the HTTP listener, handling HTTP traffic for incoming requests, and can handle HTTPS traffic, once properly configured.

- Tomcat server

  Provides a servlet container for the operation of the Administrative UI. The Apache web server communicates to the Tomcat server through a Tomcat connector named mod_jk.

You can supply CA Support with log files related to these components to troubleshoot problems in your CA SiteMinder® Federation Standalone environment.

Two Apache logs that aid CA SiteMinder® Federation Standalone troubleshooting are:

mod_jk.log

mod_jk.log is enabled by default with the product. After the first contact with the federation server, information begins logging to this file.

To modify this log file:

1. Navigate to *federation_install_dir*\secure-proxy\httpd\conf
2. Open the httpd.conf file.
3. Set the following lines in the file to reflect these settings:

   ```
   JkLogFile "path_to_mod_jk_log"
   JkLogLevel debug
   JkRequestLogFormat "%w %V %T %m %h %p %U %s"
   ```

   Note: The path "logs/mod_jk.log" is the default location for the JkLogFile the entry. Use the default or set this path to your preferred location.

To disable the mod_jk.log, comment out or remove these lines from the file.

httpclient.log

For debug purposes only, you can enable the httpclient.log. The httpclient.log file is located in *federation_install_dir*\secure-proxy\proxy-engine\logs.

To modify this log file:

1. Navigate to *federation_install_dir*\secure-proxy\proxy-engine\conf.
2. Open the server.conf file
3. Change the following line:

   ```
   httpclientlog="yes"
   ```

To modify the location of the httpclient.log file and the log level, edit the httpclientlogging.properties file. Locate this file is in the directory *federation_install_dir*\secure-proxy\Tomcat\properties.

Appendix A: Open Format Cookie Details

This section contains the following topics:

Contents of the Open Format Cookie (see page 405)

Contents of the Open Format Cookie

The federation open format cookie lets applications assert user attributes to CA SiteMinder® Federation Standalone and consume user attributes encapsulated by CA SiteMinder® Federation Standalone. The open format cookie has the following general characteristics:

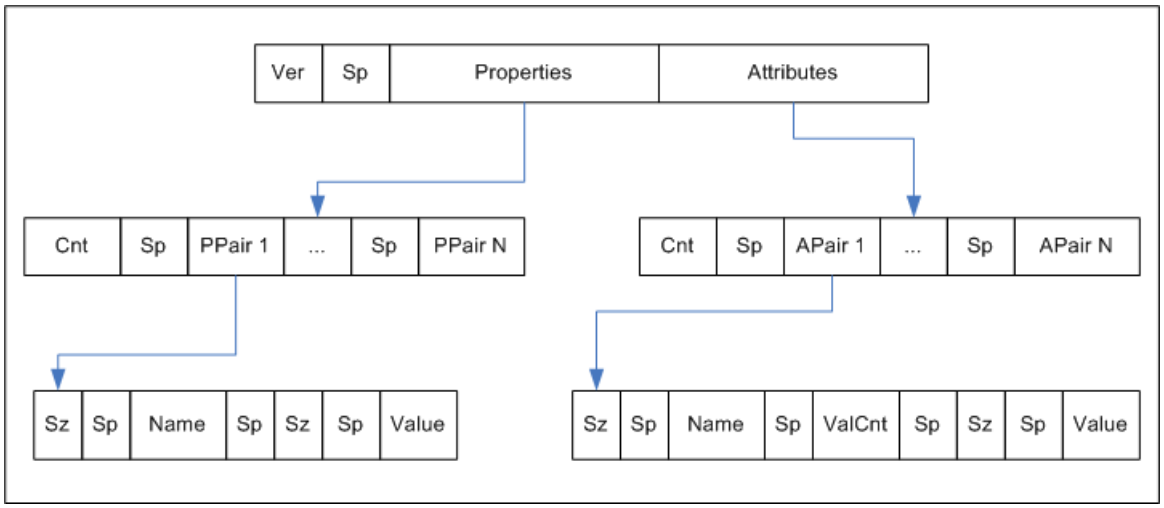
- The cookie is accessible by applications written in any programming language.
- The cookie content consists of a string of UTF-8 bytes, which supports international character sets.
- The combined size in UTF-8 bytes of each name/value pair precedes the name/value pair.
- Space characters are added for legibility.
- The cookie is simple to parse and easily extensible.

Important! If the cookie contains any unsafe characters such as '=', enclose the value in double quotes. You can specify this option through the user interface, or through the SDK.

The open format cookie contains the following property information:

- Cookie Version
- Name ID
- Name ID Format
- Session ID
- AuthnContext
- UserDN (same as User ID)
- UserConsent
- Login ID
- ExpiresON (expiration time)

The following diagram shows the open format:



Key:

- Ver — the cookie format version. This value is 1.
- Sp — an ASCII space character, used only to improve readability
- Properties — information about the principal
- Attributes — SAML attributes from the Assertion
- Cnt — the number of name value pairs that follow, represented in ASCII
- Sz — the length of the name or value that follows
- ValCnt — the number of attribute values

The Backus-Naur Form (BNF) for this format is following (0* means 0 or more; 1* means at least 1).

- DIGIT = ASCII digit (0 through 9)
- CHAR = UTF-8 character
- Sp = ASCII space (character 32)
- Token = 1*CHAR
- Cookie = Version Sp Properties Attributes
- Version = 1*DIGIT
- Cnt = 1*DIGIT
- Properties = Cnt 1*PPair
- Attributes = Cnt 0*APair

- ValCnt = 1*DIGIT
- PPair = Sz Sp Name Sp Sz Sp Value
- APair = Sz Sp Name Sp ValCnt Sp Sz Sp Value
- Sz = 1*DIGIT
- Name = Token

Value = Token

Appendix B: Encryption and Decryption Algorithms

This section contains the following topics:

Open Format Cookie Encryption Algorithms (see page 409)
Digital Signing and Private Key Algorithms (see page 410)
Back Channel Communication Algorithms (see page 410)
Backend Communication Algorithms (SPS Server) (see page 411)
Java SDK Encryption Algorithms (see page 411)
Federation System Crypto Algorithm (see page 411)
Internal Key Encryption Algorithms (see page 412)
SSL Key Algorithms for the Apache Web Server and Administrative UI (see page 412)

Open Format Cookie Encryption Algorithms

The open format cookie supports the following options for password-based encryptions:

FIPS_Compat and FIPS_Migration Modes

PBE/SHA1/AES/CBC/PKCS12PBE-1000-128

PBE/SHA1/AES/CBC/PKCS12PBE-1000-192

PBE/SHA1/AES/CBC/PKCS12PBE-1000-256

PBE/SHA256/AES/CBC/PKCS12PBE-1000-128

PBE/SHA256/AES/CBC/PKCS12PBE-1000-192

PBE/SHA256/AES/CBC/PKCS12PBE-1000-256

PBE/SHA1/3DES_EDE/CBC/PKCS12PBE-1000-3

PBE/SHA256/3DES_EDE/CBC/PKCS12PBE-1000-3

FIPS_Only Mode

AES128/CBC/PKCS5Padding

AES192/CBC/PKCS5Padding

AES256/CBC/PKCS5Padding

3DES_EDE/CBC/PKCS5Padding

Digital Signing and Private Key Algorithms

CA SiteMinder® Federation Standalone uses the following algorithms for partnership signing options.

Encryption Key Algorithms

RSA-V15, RSA-OEAP

Encryption Block Algorithms

3DES, AES-128, AES-256

CA SiteMinder® Federation Standalone uses the following algorithms for Private Key generation (Certificate/Keys):

Key Algorithm

RSA

Sign Algorithms

MD5withRSA, SHA1withRSA, SHA256withRSA & SHA512withRSA

Back Channel Communication Algorithms

For back channel communication used for HTTP-Artifact single sign-on and SAML 2.0 Single Logout, CA SiteMinder® Federation Standalone supports the following ciphers, depending upon FipsMode:

FIPS_Compat and FIPS_Migration Modes—RC4 and AES

RSA_With_RC4_SHA

RSA_With_RC4_MD5

RSA_With_AES_128_CBC_SHA

RSA_With_AES_256_CBC_SHA

FIPS_Only Mode—AES only

RSA_With_AES_128_CBC_SHA

RSA_With_AES_256_CBC_SHA

Backend Communication Algorithms (SPS Server)

For Backend Communication (SPS-backend server), following ciphers are being supported depending upon FipsMode of the setup. These are defined in <fedroot>\secure-proxy\proxy-engine\conf\server.conf.

FIPS_Compat and FIPS_Migration Modes

ciphers="-RSA_With_Null_SHA,+RSA_With_Null_MD5,-RSA_With_RC4_SHA,+RSA_With_RC4_MD5,+RSA_With_RC2_CBC_MD5,+RSA_With_DES_CBC_SHA,+RSA_With_DES_CBC_MD5,+RSA_With_3DES_EDE_CBC_MD5,+RSA_Export_With_RC4_40_MD5,-RSA_Export_With_DES_40_CBC_SHA,+RSA_Export_With_RC2_40_CBC_MD5,-DH_RSA_With_DES_CBC_SHA,-DH_RSA_With_3DES_EDE_CBC_SHA,-DH_RSA_Export_With_DES_40_CBC_SHA,-DH_DSS_With_DES_CBC_SHA,-DH_DSS_Export_With_DES_40_CBC_SHA,-DH_Anon_With_RC4_MD5,-DH_Anon_With_DES_CBC_SHA,-DH_Anon_With_3DES_EDE_CBC_SHA,-DH_Anon_Export_With_DES_40_CBC_SHA,-DH_Anon_Export_With_RC4_40_MD5,-DHE_RSA_With_DES_CBC_SHA,-DHE_RSA_Export_With_DES_40_CBC_SHA,-DHE_DSS_With_DES_CBC_SHA,-DHE_DSS_Export_With_DES_40_CBC_SHA,-Null_With_Null_Null"

FIPS_ONLY Mode

fipsciphers="+DHE_DSS_With_AES_256_CBC_SHA,
+DHE_RSA_With_AES_256_CBC_SHA, +RSA_With_AES_256_CBC_SHA,
+DH_DSS_With_AES_256_CBC_SHA, +DH_RSA_With_AES_256_CBC_SHA,
+DHE_DSS_With_AES_128_CBC_SHA, +DHE_RSA_With_AES_128_CBC_SHA,
+RSA_With_AES_128_CBC_SHA, +DH_DSS_With_AES_128_CBC_SHA,
+DH_RSA_With_AES_128_CBC_SHA, +DHE_DSS_With_3DES_EDE_CBC_SHA,
+DHE_RSA_With_3DES_EDE

Java SDK Encryption Algorithms

The CA SiteMinder® Federation Standalone Java SDK supports the following encryption algorithms:

Without a Password

"AES/CBC/PKCS5Padding"

With a Password

"PBE/SHA1/AES/CBC/PKCS12PBE-5-128"

Federation System Crypto Algorithm

FMCrypto Encryption/Decryption Algorithm

AES_128

Internal Key Encryption Algorithms

CA SiteMinder® Federation Standalone uses the following internal key encryption/decryption algorithms, depending on the FIPS mode of operation:

FIPS_MIGRATE and FIPS_ONLY Modes

AES_128

FIPS_COMPAT Mode

RC2

SSL Key Algorithms for the Apache Web Server and Administrative UI

CA SiteMinder® Federation Standalone uses the following algorithms for the embedded Apache web server SSL communication:

Apache SSL key generation

SHA1withRSA

Key encryption

DES-EDE3-CBC

CA SiteMinder® Federation Standalone uses the following algorithm for SSL communication to the Administrative UI:

SSL Key password encryption

aes-128-cbc

Index

A

Account Linking to Establish a Federated Identity • 25
Action Button for Object Lists • 32
Activate SSL on the Secondary Failover System • 333
Activate the Partnership • 54
Add a Claim to an Assertion • 170
Add a CRL to the CDS • 126
Add a Name ID to the Assertion • 45
Add an OCSP Responder to the CDS • 130
Add Session Attributes to the Assertion Configuration • 161
Add Single Logout • 60
Add the Root Certificate Authority to the Certificate Database • 78
Add the Server Certificate to the Certificate Database • 80
Administrative Session Interaction • 356
Administrative UI • 31
Administrative UI Overview • 31
Administrator Session Management • 355
Advanced User Attribute Mapping Examples • 100
Alias Attribute Use Case • 93
Aliases to Reference Certificate Data Store Content • 116
Application Integration at the Relying Party • 299
Apply Mappings to Assertion Attributes • 105
Assertion Attribute Configuration Examples • 160
Assertion Configuration • 157
Assertion Configuration at the Asserting Party • 157
Assertion Processing Customization (Relying Party) • 177
Assertion Validity for Single Sign-on • 186
Attributes for Customizing an Application • 28
Audit Logging • 385
Authenticate Users Using an OAuth Authorization Server • 214
Authentication Context Processing (SAML 2.0) • 259
Authentication Context Processing for IdP-initiated SSO • 262
Authentication Context Processing for SP-Initiated SSO • 262
Authentication Options using Partnership Federation • 185
AuthnRequest Query Parameters Used by an SP • 246

B

Back Channel Authentication for Artifact SSO • 190
Back Channel Communication Algorithms • 410
Back Channel Configuration for Single Logout • 255
Back up an Existing Configuration • 396
Backend Communication Algorithms (SPS Server) • 411
Basic SAML 2.0 Partnership • 37
Before You Configure an LDAP Connection over SSL • 76
Bit Masks in Mask Attribute Mapping • 96

C

CA SiteMinder® Connector Settings • 348
CA SiteMinder® Integration with CA SiteMinder® Federation Standalone • 285
CA Technologies Product References • 3
Certificate and Private Key Usage • 115
Certificate Authority (CA) Certificate Usage • 139
Certificate Imports • 113
Certificates for SSL Connections • 118
Certificates to Secure the Artifact Back Channel • 118
Change the Audit Log Storage Type • 387
Change the Default Administrator Password (Optional) • 354
Change the Default Administrator Password from the Command Line • 355
Change the Default Administrator Password from the UI • 354
Configuration Procedures Beyond the Simple Partnership • 67
Configure a Credential Selector Page • 220
Configure a Partnership Between the Federation System and Enterprise • 224
Configure a Policy to Generate a Session at Each Site • 289
Configure an Authentication Context Template • 264
Configure Artifact SSO at the IdP • 64
Configure Artifact SSO at the SP • 64
Configure Assertion Options • 158

Configure Attribute Mapping at the Relying Party • 307
Configure Authentication Context Processing at the IdP • 268
Configure Authentication Context Requests at the SP • 270
Configure Claims Transformation at the Asserting Party • 167
Configure ECP at the Identity Provider • 205
Configure ECP at the Service Provider • 205
Configure Federation Users • 150
Configure HTTP Headers to Pass Assertion Data (SAML only) • 302
Configure Local Account Linking (SAML 2.0) • 310
Configure Redirections to an SSL Load Balancer (optional) • 343
Configure Signature Processing at the IdP • 57
Configure Signature Processing at the SP • 59
Configure Single Logout • 253
Configure Single Logout at the IdP • 61
Configure Single Logout at the SP • 62
Configure Single Sign-on • 53
Configure Single Sign-On Between the Federation System and an Identity Provider • 223
Configure Social Sign-on • 213
Configure SSL-enabled Failover Behind a Load Balancer • 330
Configure SSL-enabled Failover Behind a Proxy Server • 333
Configure the Behavior of the Server Trace Log File • 377
Configure the Connector Settings • 292
Configure the HTTP-Artifact Back Channel • 191
Configure the IdP Partner • 39
Configure the Load Balancer • 340
Configure the Oracle Wire Protocol Driver • 87
Configure the Partnership Entities • 42
Configure the Partnership for Attribute Query Support • 194
Configure the SAML 2.0 Attribute Authority • 194
Configure the Session Store Timeout for Heavy Load Conditions • 399
Configure the SP Partner • 47
Configure the SQL Server Wire Protocol Driver • 88
Configure User Attribute Mappings • 91
Configure User Identification at the Relying Party • 155
Confirm the Authentication Mode and URL for SSO • 163
Confirm the Entity Configuration • 110, 114
Confirm the IdP-to-SP Partnership Settings • 47
Confirm the SP Partner Settings • 54
Connect to External User Stores • 352
Connect to the ODBC Directory • 41, 49
Constant Use Case • 98
Construct Attribute Mapping Rules Using the Proper Syntax • 305
Contact CA Technologies • 3
Contents of the Open Format Cookie • 405
Cookie Delegated Authentication Sample Setup • 234
Cookie Method for Passing User Identity • 229
Cookie Settings for Session and Identity Cookies • 350
Copy Key and Certificate Files from the r12 System • 369
Copy the SSL Migration Tool to Same Folder as the Key/Certificate Files • 369
Create a Common View of the Same User Information Across Directories • 89
Create a Local OAuth Client Entity • 216
Create a Partnership to Configure Single Sign-On • 218
Create a SQL Server Data Source on Windows • 388
Create a SQL Server Data Sources on UNIX Systems • 389
Create a Target Resource • 55, 66
Create a Web Page to Initiate Single Sign-on • 55
Create a Web page to Initiate Single Sign-on (Artifact) • 66
Create an Authentication Method Group • 223
Create an Oracle Data Source on a UNIX System • 391
Create an Oracle Data Source on Windows • 390
Create or Modify the Remote Entity of an Authorization Server • 217
Create the Certificate Database Files • 77
Create the IdP-to-SP Partnership • 44
Create the SP-to-IdP Partnership • 52
Customize a User Consent Form (Optional) • 202
Customize Assertion Content • 173
Customize Assertion Processing (Relying Party) • 177
Customize the Auto-POST form for HTTP-POST SSO • 184
Customize the Header and Footer on the Credential Selector Page • 225

D

Deactivate SSL • 363
Delegated Authentication • 227
Delegated Authentication Configuration • 234
Delegated Authentication Overview • 227
Delete a Claim from an Assertion • 171
Delivery of Assertion Data to the Provisioning Application • 313
Deploy a Message Consumer Plug-in • 180
Deploy an Assertion Generator Plug-in • 174
Deployment Modes and FIPS Settings • 346
Deployment Settings • 346
Detailed Local Entity Configuration • 108
Detailed Remote Entity Configuration • 109
Determine Authentication Context and Strength Levels with your Partner • 264
Determine how a User Authenticated at an Identity Provider • 260
Determine which Session Attributes are Available • 161
Digital Signing and Private Key Algorithms • 410
Disable Signature Processing • 46, 54
Disable UI Administration • 357
Documentation Changes • 4
Dynamic Provisioning of a User Identity at the Relying Party • 308

E

Editing Entities from the Partnership • 146
Employ AllowCreate for User Identification (SAML 2.0) • 156
Enable Authentication Context Requests at the SP • 271
Enable OCSP Status Checks • 131
Enable Signature Processing • 56
Enable the Assertion Generator Plug-in • 175
Enable the Connector at the Partnership Level • 293
Enable the HTTP POST Binding at the IdP • 210
Enable the HTTP POST Binding at the SP • 211
Enable the Message Consumer Plug-in in the UI • 179
Enable the Session Store • 70
Enable the System to Serve as an Attribute Authority (IdP->SP) • 197
Enable the System to Serve as an Attribute Requester (SP->IdP) • 198
Enable User Consent at the IdP • 201
Enable WSFED Metadata Exchange • 323
Enable WSFED Sign-Out • 257
Encryption and Decryption Algorithms • 409
Encryption and Decryption Operations • 118
Encryption Configuration at a SAML 2.0 IdP • 276
Encryption Configuration at a SAML 2.0 SP • 279
Enforcing the One Time Use of an Assertion • 295
Enhanced Client or Proxy Profile Overview (SAML 2.0) • 203
Entity Configuration Changes from a Partnership • 111
Entity Type Choice • 107
Entity-level Metadata Export • 320
Environments that Require a Shared Session Store • 71
Establish a Connection to the Certificate Database • 83
Establish a User Directory Connection • 40, 47
Establish Connections to User Directories • 90
Examine Secure Proxy Engine Logs to Troubleshoot the System • 402
Export Certificates from the CDS using the Administrative UI • 137
Export Metadata to Aid Partnership Configuration • 319
Exporting a Partnership • 147
Expression Use Case • 99

F

Failed Authentication Handling Using Redirect URLs (Relying Party) • 316
Failover Introduction • 325
Failover Support for Federation System • 325
Federated User Identification for a Partnership • 149
Federation Data Object Trace Logging • 384
Federation Data Stored in the Session Store • 69
Federation Entity Configuration • 107
Federation in Your Enterprise • 23
Federation Logging Overview • 373
Federation Profile for Single Sign-on • 29
Federation System Administration • 345
Federation System Crypto Algorithm • 411
Federation Users Configuration at the Asserting Party • 149
Federation Web Services (FWS) Logging • 374
Filtering Object Lists • 33
FIPS 140-2 Support Offered by CA SiteMinder® Federation Standalone • 19
ForceAuthn and IsPassive Processing at the IdP • 243

G

Generate a Certificate Request • 122
Generate a New Certificate Signing Request • 124
Generate a New Key/Certificate Pair Using the UI or a Third-party Tool • 134
Getting Started with a Simple Partnership • 37
Give the Metadata Exchange URL to Your Partner • 323
Group Name Use Case • 94

H

How the Third Party WAM Passes the User Identity • 228
How To Add Session Attributes to an Assertion • 160
How to Configure Claims Transformation at the Asserting Party • 163
How to Configure Failover • 327
How to Configure Failover with SSL Enabled • 329
How to Configure Federation System Administrators • 351
How to Configure Load Balancing • 337
How to Configure SAML 2.0 HTTP POST Binding • 209
How to Connect to an LDAP User Directory Over SSL • 76
How to Create an Entity by Importing Metadata • 111
How to Create an Entity without Using Metadata • 107
How to Enable SAML 2.0 Attribute Query Support • 192
How to Enable SSL for the Apache Web Server and the UI • 360
How To Enable WS-Federation Metadata Exchange • 321
How To Follow a Single Transaction in a Log • 393
How to Generate a Key/Certificate Pair • 122
How to Get User Consent to Send an Assertion • 199
How to Integrate CA SiteMinder® Federation Standalone and CA SiteMinder® • 285
How to Manage the Authentication Session Duration at a Service Provider • 281
How to Migrate SSL Keys and Certificates • 367
How To Restore a System to a Previous Configuration • 395
How to Retrieve User Attribute Values from a Third-Party Source • 195
How to Send Certificates to Your Partner • 132
How to Verify that Certificates are Valid Using CRLs • 124
How to Verify that Certificates are Valid using OCSP • 129
HTTP Header Protection for a Proxy Mode Deployment at the Relying Party • 347

I

Identify the Partnership Entities • 50
Identity Mapping to Establish a Federated Identity • 26
IDP Discovery Configuration at the Identity Provider • 206
IDP Discovery Configuration at the Service Provider • 207
IDP Discovery Profile (SAML 2.0) • 206
IdP-initiated SSO (SAML 2.0 Artifact or POST) • 241
Implement the AssertionGeneratorPlugin Interface • 174
Implement the MessageConsumerPlugin Interface • 178
Import a CA Certificate • 139
Import a Key/Certificate Pair from an Existing File • 120, 135
Import a Signed Certificate Response • 123, 136
Import the Key/Cert Pair into the CDS • 135
Include a Session Duration Attribute in an Assertion • 282
Integrate with CA SiteMinder® using the SiteMinder Connector • 287
Intended Audience • 21
Internal Key Encryption Algorithms • 412
Introduction • 17
IP-initiated Single Sign-on (WSFED) • 248

J

Java SDK Encryption Algorithms • 411

K

Key and Certificate Management • 115

L

LDAP Directory Connection • 74
Learn the Attribute Expression Guidelines • 165
Links to Servlets which Initiate Single Sign-on • 239
Load Balancing and Failover for LDAP User Directories • 74
Load Balancing Support for Federation System • 337

Local Account Linking for Provisioning • 309
Local Logout at the SP (SAML 2.0) • 258
Log in to the Administrative UI • 34
Log Settings • 381
log4j.properties File for server.log • 383
Logging Out of User Sessions • 251
Logs to Monitor Federation Activities • 373

M

Maintain the Same Configuration for Each System • 334
Manage Certificate Cache Refresh and Grace Period • 128, 131
Managing Single Logout Across a Network Using HTTP-Redirect and SOAP • 252
Map a First Name Attribute with an Alias Mapping Type • 100
Map a Last Name Attribute with an Alias Mapping Type • 101
Map a Sort Name Attribute with Expression and Alias Mapping Types • 102
Map Customers with Group and Constant Mapping Types • 103
Map the Account Status with the Mask and Expression Mapping Types • 104
Mapping Assertion Attributes to Application Attributes (SAML Only) • 302
Mask Use Case • 95
Metadata Exchange Transaction Flow • 322
Metadata Export Overview • 319
Metadata File Selection • 112
Methods to Create an Entity • 107
Migrate an OAuth Authentication Scheme Set-up to OAuth Partnership • 219
Migrate or Export SSL Keys and Certificates • 369
Migrate the SSL Setup to the Secondary System • 331
Modify and Delete Mappings • 304
Modify the System Settings • 345

N

New Object Creation • 32

O

Object Management • 32
Objects Lists • 32
Obtain a Key/Certificate Pair for Federated Transactions • 119
OCSP Prerequisites • 130
ODBC Data Source on Solaris Configuration Requirement • 87
ODBC Directory Connection • 85
ODBC Directory Failover Configuration • 85
Open Format Cookie Details • 405
Open Format Cookie Encryption Algorithms • 409
Overview • 17

P

Page Displays • 34
Partnership Activation • 147
Partnership Confirmation • 146
Partnership Creation • 143
Partnership Creation and Activation • 143
Partnership Definition • 144
Partnership Identification and Configuration • 145
Partnership-Level Metadata Export • 320
Prerequisites for Claims Transformation • 165
Producer-initiated SSO (SAML 1.1) • 239
Product Components • 18
Programmerless Federation • 20
Protecting a Federated Network Against Cross-Site Scripting • 296
Protecting Federated Communication • 295
Protection Level Assignments for a Context Template • 266
Proxied Attribute Query Overview • 196
Proxy Engine Hangs and Stops Processing Requests • 400

Q

Query String Delegated Authentication Sample Setup • 235
Query String Method for Passing User Identity • 231

R

Reactivate SSL • 365
Redirect a User to the Target Application • 299
Relying Party Interaction with Applications • 299
Remote Provisioning • 312
Remote Provisioning Configuration • 315
Remove SSL from the Embedded Apache Server and the UI • 366
Replace or Resubmit a Certificate Signing Request for SSL • 366
Request an SSL Server Certificate • 361
Require User Consent at the SP • 203

Resolving Signature Verification Failures • 401
Restore Federation System Configuration • 395
Revert to a Backed-up Configuration • 396
RP-initiated Single Sign-on (WSFED) • 249

S

Sample Federation Network • 38
Secure a Federated Environment • 295
Securing Connections Across the Federated Environment • 296
Securing the IdP Discovery Target Against Attacks • 208
Select an Entity to Import • 113
Select Users as Administrators • 353
Send the Certificate File to your Partner • 138
Server Status Monitoring • 345
Server Trace Logging • 376
server.log File Setup • 379
Session Attribute Examples in the Administrative UI • 162
Session Duration Management at a Service Provider • 281
Session Validity at a Service Provider • 188
Set the Audit Log Name and Location (Optional) • 386
Set up an Authentication Context Template • 265
Set up Failover at Each Federation System • 327
Set Up Single Sign-on • 46
Set Up the Artifact Profile for SSO • 63
Set up the Federation Systems to Work with a Load Balancer • 341
Set up the Proxy Server or Load Balancer for Failover • 329, 334
Set up the Sample Users for the Data Source • 40, 48
Set Up the Server Trace Log Configuration File • 376
Sign and Encrypt Federation Messages • 273
Signature Configuration at a SAML 1.1 Producer and WSFED IP • 273
Signature Configuration at a SAML 2.0 IdP • 275
Signature Configuration at a SAML 2.0 SP • 277
Signature Verification at a SAML 1.1 Consumer and a WSFED RP • 274
Signing and Verification Operations • 117
Sign-Out Overview (WS-Federation) • 257
Single Logout (SAML 2.0) • 251
Single Sign-on Configuration • 181
Single Sign-on Configuration (Asserting Party) • 181
Single Sign-on Configuration (Relying Party) • 185
Single Sign-on Initiation (SAML 2.0) • 189
Specify Federation Users for Assertion Generation • 45
Specify How the IdP Obtains the Authentication Context • 268
Specify the User Identification Attribute • 53
SP-initiated SSO (SAML 2.0) • 244
SSL Administration for Federation System • 359
SSL Administration for the Apache Web Server and the UI • 359
SSL Key Algorithms for the Apache Web Server and Administrative UI • 412
SSL Migration Tool Command Arguments • 370
SSL-enable the LDAP User Directory Connection • 82
Status Redirects for HTTP Errors (SAML 2.0 IdP) • 189
Storing User Session, Assertion, and Expiry Data • 69
System Performance Troubleshooting • 399

T

Terminology Used in this Guide • 21
Test a User Directory Connection from the Directory List • 89
Test Artifact Single Sign-on • 66
Test POST Single Sign-on • 56
Test Single Logout • 63
Test the Partnership (Artifact SSO) • 65
Test the Partnership (POST Profile) • 55
The Partnership Model • 29
Third-party WAM Configuration for Cookie Delegated Authentication • 237
Third-party WAM Configuration for Query String Delegated Authentication • 238
Transaction IDs to Aid Federation Troubleshooting • 392
Transform a Claim in an Assertion • 168
Troubleshoot Certificate Signature Verification for Back Channel Communication • 141
Troubleshoot the SSL Connection to the LDAP User Directory • 84
Troubleshooting • 399
Two SSO Transactions Fail Using the Same Browser Session • 402

U

UI Login Password Conditions with Active Directory • 35

Understanding Skew Time for SLO Request Validity • 253
Unsolicited Response Query Parameters Used by the IdP • 242
Update a CRL • 127
Update Certificates in the Certificate Data Store • 138
Upload the Signed Server Certificate • 362
URLs to Initiate Single Sign-on • 239
Use an ODBC Database for Audit Logging (Optional) • 387
User Consent Example • 201
User Directory Connections for Authentication • 73
User Directory Management Overview • 73
User Identification Across the Partnership • 25
User Identification at the Relying Party • 153
User Provisioning to Establish a Federated Identity • 27
Using HTTP Headers to Pass Assertion Data (SAML only) • 301
Using the Application Attributes Definitions Table • 303

V

Verify that the Certificates are in the Database • 81
Verify the Prerequisites • 216
Verify the SSL Connection to the LDAP Directory • 84

W

Wizards for Configuring Objects • 34