

CA SiteMinder®

Federation Security Services Guide

r12.0 SP3



Fifth Edition

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 may not be copied, transferred, reproduced, disclosed, modified or duplicated, in whole or in part, without the prior written consent of CA. This Documentation is confidential and proprietary information of CA and may not be disclosed by you or used for any purpose other than as may be permitted in (i) a separate agreement between you and CA governing your use of the CA software to which the Documentation relates; or (ii) a separate confidentiality agreement between you and CA.

Notwithstanding the foregoing, 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 © 2012 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[®]
- CA SiteMinder[®] SAML Affiliate Agent
- CA SiteMinder[®] Web Agent Option Pack
- CA SiteMinder[®] for Secure Proxy Server (CA SPS)

Contact CA Technologies

Contact CA Support

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

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

Providing Feedback About Product Documentation

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

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

Documentation Changes

The following documentation updates have been made since the fourth edition of this documentation:

- [Specify the Maximum Length of Assertion Attributes](#) (see page 247)—Added under the topic Configure Attributes to Include in SAML 1.x Assertions (Optional) (151642, 151887).
- [Specify the Maximum Length of Assertion Attributes](#) (see page 247)—Added under the topic Configure Attributes for Assertions (optional) (151642, 151887).
- [Specify the Maximum Length of Assertion Attributes](#) (see page 247)—Added under the topic Configure Attributes for WS-Federation Assertions (optional) (151642, 151887).

Contents

Chapter 1: Federation Security Services Overview 19

Introduction to SiteMinder Federation Security Services	19
Terminology for Partners in a Federation	20
SAML Profiles Supported by SiteMinder	20
WS-Federation	20
User Mapping.....	21
SiteMinder Components for Federation Security Services	22
SAML Assertion Generator.....	22
WS-Federation Assertion Generator.....	23
SAML and WS-Federation Authentication Schemes	24
Federation Web Services Application	24
SAML Affiliate Agent	27
Secure Proxy Server Federation Gateway.....	28
Debugging Features	28
APIs for Federation Security Services.....	29
Internationalization in Federation Security Services	30
Federated Single Sign-on with Security Zones	30
Federation Use Cases.....	31
Use Case 1: Single Sign-on Based on Account Linking.....	32
Use Case 2: Single Sign-on Based on User Attribute Profiles	40
Use Case 3: Single Sign-on with No Local User Account	42
Use Case 4: Extended Networks	45
Use Case 5: Single Logout	48
Use Case 6: WS-Federation Signout.....	51
Use Case 7: Identity Provider Discovery Profile	54
Use Case 8: Multi-protocol Support.....	58
Use Case 9: SAML 2.0 User Authorization Based on a User Attribute	61
Use Case 10: SAML 2.0 Single Sign-on with No Name ID at the IdP.....	63
Use Case 11: SAML Artifact SSO Using Security Zones.....	66
Use Case 12: SAML 2.0 SSO Using Attributes from a Web Application.....	70
Use Case 13: SSO with Dynamic Account Linking at the SP	75
SiteMinder Administrative User Interfaces.....	79

Chapter 2: Deploy Federation Using the Sample Application 81

Legacy Federation Sample Application Overview	81
Legacy Sample Application Deployment	81

Sample Application Components	82
Prerequisites to Deploy the Sample Application	84
How To Run the Sample Application	85
Set the Path Variable on the Policy Server System	85
Configure a Web Agent in the FSS Administrative UI	86
Modify the FederationSample.conf File	86
Modify the SetupFederationSample.pl Script (Optional)	88
Run the Sample Application on the Policy Server System	89
Set up the Web Agent System	90
Test Single Sign-on with the Sample Application	92
Test Single Logout with the Sample Application	93
Review Application-Generated SiteMinder Objects	94

Chapter 3: Deploy Federation Using a Manual Configuration **95**

Manual SiteMinder-to-SiteMinder Deployment Overview	95
Manual Deployment Prerequisites	96
Sample Federation Network	96
Identity Provider Data for a Basic Configuration	98
Identity Provider Data for an Advanced Configuration	99
Service Provider Data for a Basic Configuration	100
Service Provider Data for an Advanced Configuration	101
Set Up the Identity Provider	102
Install the IdP Policy Server	102
Point the Policy Server to the IdP LDAP Policy Store	103
Set Up the IdP User Store	104
Enable Policy Server Trace Logging at the IdP	105
Install the IdP Web Agent	105
Install the IdP Web Agent Option Pack	106
Configure the Web Server with the Web Agent Option Pack	106
Enable Web Agent Option Pack Logging at the IdP	110
Specify the User Store for the IdP Policy Server	110
Set up an Affiliate Domain at the IdP	112
Add the Service Provider to the Affiliate Domain at the IdP	113
Select Users for which the IdP Generates Assertions	115
Configure a Name ID for Inclusion in the Assertion	116
Identify the SP, IdP, and Other General Settings	116
Configure POST Single Sign-on at the IdP	117
Federation Web Services Access	118
Configure the Service Provider	118
Set Up the Service Provider	118
Install the SP Policy Server	119

Point the Policy Server to the SP LDAP Policy Store.....	120
Set Up the SP User Store.....	120
Enable Trace Logging for Federation Components at the SP	121
Install the SP Web Agent.....	122
Install the SP Web Agent Option Pack	122
Configure the Web Server with the Web Agent Option Pack	122
Enable Web Agent Option Pack Logging at the SP	125
Specify the User Store for the SP Policy Server.....	126
Specify the POST Binding Authentication at the SP	127
Configure the SAML 2.0 Authentication Scheme at the SP.....	128
Protect the Target Resource at the SP	129
Test SAML 2.0 Single Sign-on	131
Add Functionality to the Federation Deployment	134
Configure Single Logout	134
Configure SAML 2.0 Artifact Single Sign-on	137
Include an Attribute in the Assertion.....	144
Configure Digital Signing (required for POST Binding).....	145
Encrypt and Decrypt the Assertion	148

Chapter 4: Overview of a SiteMinder Federation Setup 151

Installation Overview	151
Conventions in the Installation Overview Procedures.....	152
Set Up Asserting Party Components	153
Install the Policy Server at the Asserting Party	154
Set up Affiliate Domains and Add Sites to these Domains.....	154
Install a Web Agent or SPS Federation Gateway at the Asserting Party	155
Install an Application Server for the Web Agent Option Pack (Asserting Party).....	156
Install the Asserting Party Web Agent Option Pack.....	156
Configure Federation Web Services (Asserting Party)	157
Allow Access to Federation Web Services (asserting party)	158
Enable the Signing of SAML POST Responses	158
Create Links to Target Resources (optional)	158
Set Up Relying Party Components	161
Install the Relying Party Policy Server.....	162
Configure a SAML or WS-Federation Authentication Scheme.....	163
Protect Target Resources at the Relying Party.....	163
Install a Web Agent or SPS Federation Gateway (Relying Party)	164
Install a Web or Application Server for the Web Agent Option Pack (Relying Party)	164
Install the Web Agent Option Pack at the Relying Party	165
Configure Federation Web Services at the Relying Party	165
Allow Access to Federation Web Services (Relying Party)	166

Set-up the smkeydatabase for Artifact Single Sign-on (optional)	167
Create Links to Initiate Single Sign-on (optional)	167
Chapter 5: Setup the SAML 1.x Assertion Generator File	171
SAML 1.x Assertion Generator Properties File	171
Configure the SAML 1.x AMAssertionGenerator.properties File	171
Chapter 6: Review the JVMOptions File Which Creates a JVM	172
The JVMOptions.txt File	172
Chapter 7: Storing User Session, Assertion, and Expiry Data	175
Federation Data Stored in the Session Store	175
Enable the Session Store	176
Environments that Require a Shared Session Store	177
Chapter 8: Grant Access to Federation Web Services	179
Policies that Protect Federation Web Services	179
Features Associated with FWS Policies	180
Enforce the Policies that Protect Federation Web Services	181
Chapter 9: Signing and Encrypting Messages to Secure Federated Transactions	183
Certificate and Private Key Usage for Federation	183
Signing and Verification Operations	184
Encryption/Decryption Operation	185
Certificates for SSL Connections	185
Client Certificate Authentication Across the Back Channel	185
Certificates To Secure the Artifact Back Channel	185
SmKeyDatabase Overview	186
Role of the Smkeydatabase at the Asserting Party	189
Role of the Smkeydatabase at the Relying Party	189
Certificates Stored in the smkeydatabase Only at the Asserting Party	190
Certificate Revocation Lists in the smkeydatabase	191
Formats Supported by the Smkeydatabase	192
Properties File for the Key Database	192
DBLocation Setting	193
NativeDBName Setting	193
XMLDocumentOpsImplementation Setting	193
AffiliateIXMLSignatureImplementation Setting	194

IXMLSignatureImplementation Setting	194
EncryptedPassword Setting	194
IXMLEncryptDecryptImplementation Setting	194
DBUpdateFrequencyMinutes Setting	195
LDAPAccessTimeout.....	195
Modify the Key Database Using smkeytool.....	195
Smkeytool Command Syntax and Options.....	197
Smkeytool Examples for Windows Platforms	205
Smkeytool Examples for UNIX Platforms	206
Migrate AM.keystore and Update smkeydatabase.....	208
Considerations Before Migrating Key Databases	210
How To Migrate the Key Databases	211

Chapter 10: Securing a Federated Environment **215**

Protecting Federated Communication	215
Setting a One Time Use Condition for an Assertion.....	215
Securing Connections Across the Federated Environment.....	216
Protecting Against Cross-Site Scripting	217

Chapter 11: Creating Affiliate Domains **219**

Affiliate Domain Overview	219
Configure an Affiliate Domain	219
Add a Domain Object	220
Assign User Directories	220
Assign an Administrator	221
Add Entities to an Affiliate Domain	222

Chapter 12: Configure SiteMinder as a SAML 1.x Producer **223**

Prerequisites for a SiteMinder Asserting Party	223
How To Configure SiteMinder to Act as a SAML 1.x Producer	224
Optional Configuration Tasks at a 1.x Producer.....	225
Add a Consumer to an Affiliate Domain	225
Authenticate Users with No SiteMinder Session (SAML 1.x)	227
Create a Policy to Protect the Authentication URL	227
Select Users for Which the Producer Generates Assertions	228
Adding Users and Groups for Access to a Consumer	229
Excluding a User or Group from Access to a Consumer.....	229
Allowing Nested Groups Access to Consumers.....	230
Adding Users by Manual Entry.....	230
Configure a SAML 1.x Assertion	231

A Security Issue Regarding SAML 1.x Assertions.....	232
Assertion Validity for Single Sign-on	232
Generate an Assertion for One Time Use	233
Grant Access to the Service for Assertion Retrieval (Artifact SSO)	234
Add a Web Agent to the Federation Agent Group.....	234
Add Relying Partners to the FWS Policy for Obtaining Assertions (Artifact SSO)	235
Verify Basic Protection of the Assertion Retrieval Service.....	236
Configure the Authentication Scheme that Protects the Artifact Service	236
Basic Authentication to Protect the Service that Retrieves Assertions	237
Basic over SSL to Protect the Assertion Retrieval Service.....	237
Client Certificate Auth to Protect the Service that Retrieves Assertion	238
Setting Up Sessions for a SAML Affiliate Agent Consumer (optional)	241
Configure a Default or Active Session Model.....	242
Configure a Shared Session Model.....	242
Configure Attributes to Include in SAML 1.x Assertions (Optional)	243
Attribute Types	244
Configure Attributes for SAML 1.x Assertions.....	245
Specify the Maximum Length of Assertion Attributes	247
Use a Script to Create A New Response Attribute	248
Configure IP Address Restrictions for 1.x Consumers (optional)	248
Configure Time Restrictions for 1.x Consumers (optional)	248
Customize the SAML 1.x Assertion Response (optional).....	249
Implement the AssertionGeneratorPlugin Interface	249
Deploy the Assertion Generator Plug-in	250
Enable the Assertion Generator Plug-in.....	250
Creating Links to Consumer Resources for Single Sign-on	251
Choosing Whether to Protect the Intersite Transfer URL.....	253

Chapter 13: Configure SiteMinder as a SAML 1.x Consumer **255**

SAML 1.x Authentication Scheme Prerequisites	255
How To Configure SiteMinder as a SAML 1.x Consumer.....	256
Optional Tasks to Configure a SiteMinder Consumer	256
SAML 1.x Authentication Schemes.....	256
SAML 1.x Artifact Authentication Scheme Overview.....	258
SAML 1.x POST Profile Authentication Scheme Overview	260
Configure SAML 1.x Artifact Authentication	261
Configure the SAML 1.x Artifact Scheme Setup	261
Create a Custom SAML Artifact Authentication Scheme (Optional).....	263
Backchannel Configuration for HTTP-Artifact SSO.....	263
Configure SAML 1.x POST Profile Authentication	263
Create the SAML 1.x POST Common Setup and Scheme Setup.....	264

Configure a Custom SAML 1.x POST Authentication Scheme	265
Customize Assertion Processing with the Message Consumer Plug-in	265
Implement the MessageConsumerPlugin Interface	266
Deploy a Message Consumer Plug-in	267
Enable the Message Consumer Plug-in for SAML 1.x	268
Redirect Users After Failed SAML 1.x Authentication Attempts	269
Supply SAML Attributes as HTTP Headers	270
Use Case for SAML Attributes As HTTP Headers	270
Configuration Overview to Supply Attributes as HTTP Headers	272
Set the Redirect Mode to Store SAML Attributes	272
Create an Authorization Rule to Validate Users	273
Configure a Response to Send Attributes as HTTP Headers	274
Create a Policy to Implement Attributes as HTTP Headers	275
Enable Client Certificate Authentication for the Back Channel(optional)	275
Add a Client Certificate to smkeydatabase	276
Select the Client Cert Option for Authentication	277
How To Protect a Resource with a SAML 1.x Authentication Scheme	277
Configure a Unique Realm for Each SAML Authentication Scheme	278
Configure a Single Target Realm for All Authentication Schemes	279

Chapter 14: Configure SiteMinder as a SAML 2.0 Identity Provider **285**

Prerequisites for a SiteMinder Asserting Party	285
Configuration Checklist at the Identity Provider	286
How to Configure a SiteMinder Identity Provider	286
Optional Configuration Tasks for Identifying a Service Provider	287
Add a SAML 2.0 Service Provider to an Affiliate Domain	287
Select Users For Which Assertions Will Be Generated	288
Exclude a User or Group from Service Provider Access	289
Allow Nested LDAP Groups Service Provider Access	289
Add Users by Manual Entry for Access to a Service Provider	290
Specify Name Identifiers for SAML 2.0 Assertions	290
Configure a Name ID	291
Configure a SAML 2.0 Affiliation (Optional)	291
Configure Required General Information	292
Set a Password for SAML Artifact Back Channel Authentication	293
WebLogic Configuration Required for Back Channel Authentication	293
Determine Digital Signing Options	294
Validate Signed AuthnRequests and SLO Requests/Responses	295
Authentication Users with no SiteMinder Session (SAML 2.0)	296
Create a Policy to Protect the Authentication URL	297
Configure Single Sign-on for SAML 2.0	298

Assertion Validity for Single Sign-on	298
Configure an Assertion for One Time Use.....	300
Customize the Session Duration in the Assertion	301
Grant Access to the Service for Assertion Retrieval (Artifact SSO)	302
Define Indexed Endpoints for Different Single Sign-on Bindings	305
Enforce the Authentication Scheme Protection Level for SSO	309
Determine Digital Signing Options	309
Allow the Identity Provider to Assign a Value for the NameID	310
Configure IP Address Restrictions for Service Providers (optional)	311
Configure Time Restrictions for Service Provider Availability (optional)	312
Enhanced Client or Proxy Profile Overview (SAML 2.0)	312
Configure the Authentication Scheme that Protects the Artifact Service	314
Basic Authentication to Protect the Service that Retrieves Assertions	315
Basic over SSL to Protect the Assertion Retrieval Service	315
Client Certificate Auth to Protect the Service that Retrieves Assertion	316
Configure Attributes for Assertions (optional).....	319
Attributes for SSO and Attribute Query Requests	320
Configure Attributes for SSO Assertions	320
Specify the Maximum Length of Assertion Attributes	322
Set Up Links at the IdP or SP to Initiate Single Sign-on	323
Identity Provider-initiated SSO (POST or artifact binding)	324
Service Provider-initiated SSO (POST or artifact binding)	326
Configure Single Logout (optional).....	331
Single Logout Request Validity	332
Guidelines for the Single Logout Confirmation Page	333
Configure Identity Provider Discovery at the IdP	334
Enable Identity Provider Discovery Profile (optional)	334
Securing the IdP Discovery Target Against Attacks	335
Encrypt a NameID and an Assertion.....	336
Enabling Encryption	336
Request Processing with a Proxy Server at the IdP.....	337
Configure Request Processing with a Proxy Server.....	338
HTTP Error Handling at the IdP	339
Customize a SAML Response Element (optional).....	339
Implement the AssertionGeneratorPlugin Interface	340
Deploy the Assertion Generator Plug-in	341
Enable the Assertion Generator Plug-in (SAML 2.0)	341

Chapter 15: Configure SAML 2.0 Affiliations At the Identity Provider **343**

Affiliation Overview.....	343
Affiliations for Single Sign-On.....	343

Affiliations for Single Logout	344
Configure Affiliations.....	344
Assign Name IDs to Affiliations	344
Specify Users for Disambiguation for SAML Affiliations	345
View a List of Service Providers in an Affiliation	346
View Authentication Schemes That Use an Affiliation.....	346

Chapter 16: Configure SiteMinder as a SAML 2.0 Service Provider **347**

SiteMinder as a Service Provider.....	347
SAML Authentication Request Process.....	349
SAML 1.x Authentication Scheme Prerequisites	350
How to Configure a SiteMinder Service Provider.....	351
Optional Tasks to Configure a Service Provider	351
Configure the SAML 2.0 Authentication Scheme	352
Digital Signing Options at the Service Provider.....	353
Create a Custom SAML 2.0 Authentication Scheme (optional)	354
Look Up User Records for SAML 2.0 Authentication.....	354
Use a SAML Affiliation to Locate a User Record (Optional)	355
Configure Disambiguation Locally as Part of the Authentication Scheme.....	355
Configure Single Sign-on at the SP	357
Configure the Backchannel for HTTP-Artifact SSO	359
Enforcing a Single Use Policy to Enhance Security	359
Permit the Creation of a Name Identifier for SSO.....	361
Enhanced Client or Proxy Profile Overview (SAML 2.0)	362
Enable Single Logout	364
Bindings for Single Logout.....	364
Configure Single Logout	364
Enforce Assertion Encryption Requirements for Single Sign-on	365
Set Up Encryption for SSO.....	365
Supply SAML Attributes as HTTP Headers.....	366
Use Case for SAML Attributes As HTTP Headers.....	366
Configuration Overview to Supply Attributes as HTTP Headers	368
Set the Redirect Mode to Store SAML Attributes	368
Create an Authorization Rule to Validate Users.....	369
Configure a Response to Send Attributes as HTTP Headers	370
Create a Policy to Implement Attributes as HTTP Headers.....	371
IDP Discovery Configuration at the Service Provider	371
Securing the IdP Discovery Target Against Attacks.....	372
Customize Assertion Processing with the Message Consumer Plug-in.....	373
Implement the MessageConsumerPlugin Interface.....	374
Deploy a Message Consumer Plug-in.....	375

Enable the Message Consumer Plug-in (SAML 2.0)	376
Specify Redirect URLs for Failed SAML 2.0 Authentication	376
HTTP Error Handling for SAML 2.0 Authentication	378
Request Processing with a Proxy Server at the SP	379
Configure Request Processing with a Proxy Server at the SP	379
Enable Client Certificate Authentication for the Back Channel(optional)	380
Configure the Client Certificate Authentication at the Relying Party	381
Protect the Artifact Resolution Service at the Identity Provider	382
How To Protect Resources with a SAML 2.0 Authentication Scheme.....	382
Configure a Unique Realm for Each SAML Authentication Scheme.....	383
Configure a Single Target Realm for All Authentication Schemes	384

Chapter 17: Authorize Users with Attributes from an Assertion Query 389

Perform Authorizations with an Attribute Authority	389
Flow Diagram for Authorizing a User with User Attributes.....	392
How to Configure an Attribute Authority and a SAML Requester	393
Set up the Attribute Authority	393
Configure Attributes at the Attribute Authority	395
Configure the Back Channel for the Attribute Authority	395
Set up a SAML Requestor to Generate Attribute Queries.....	396
Enable Attribute Queries and Specify Attributes	397
Configure the NameID for the Attribute Query	398
Configure the Backchannel for the Attribute Query	398
Create a Federation Attribute Variable	399
Create a Policy Expression with the Federation Attribute Variable	399

Chapter 18: Configure SiteMinder as an Account Partner 401

Prerequisites for a SiteMinder Asserting Party	401
How to Configure a SiteMinder Account Partner.....	402
Optional Configuration Tasks for a SiteMinder Account Partner.....	403
Add a Resource Partner to an Affiliate Domain	403
Authenticate Users without a SiteMinder Session.....	404
Create a Policy to Protect the Authentication URL	405
Select Users for Which Assertions Will Be Generated	406
Excluding a User or Group from Resource Partner Access	407
Allow Nested LDAP Groups Resource Partner Access.....	407
Add Users by Manual Entry for Resource Partner Access.....	408
Specify Name IDs for WS-Federation Assertions	409
Configure a Name ID for a WS-Federation Assertion.....	409
Configure Required General Information for WS-Federation	409
Set the Skew Time WS-Federation Single Sign-on	410

Configure Single Sign-on for WS-Federation	411
Set the Authentication Scheme Protection Level	412
Specify IP Address Restrictions for Resource Partners (optional)	412
Set up Time Restrictions for Resource Partner Availability (optional)	413
Customize a WS-Federation Assertion (optional)	414
Implement the AssertionGeneratorPlugin Interface	414
Deploy the Assertion Generator Plug-in	415
Enable the Assertion Generator Plug-in (WS-Federation)	415
Configure Attributes for WS-Federation Assertions (optional)	416
Configure Assertion Attributes for WS-Federation	417
Specify the Maximum Length of Assertion Attributes	419
Use a Script to Create a New Attribute	420
Configure Signout for WS-Federation	420
Enable Signout	421
Validate Signout Requests that are Digitally Signed	421
Set Up Links to Initiate WS-Federation Single Sign-on	422
Initiate Single Sign-on at the Account Partner	422
Initiate Single Sign-on at the Resource Partner	423

Chapter 19: Configure SiteMinder as a Resource Partner 425

SAML 1.x Authentication Scheme Prerequisites	425
How to Configure a SiteMinder Resource Partner	426
Optional Tasks to Configure a SiteMinder Resource Partner	427
WS-Federation Authentication Scheme Overview	427
Configure the WS-Federation Authentication Scheme	428
Locate User Records for Authentication	429
Obtain a LoginID for a WS-Federation User	430
Use a Search Specification to Locate a WS-Federation User	431
Configure WS-Federation Single Sign-on at the Resource Partner	431
Create a Custom WS-Federation Authentication Scheme (optional)	432
Implement WS-Federation Signout	433
Enable Signout	433
Customize Assertion Processing with the Message Consumer Plug-in	434
Implement the MessageConsumerPlugin Interface	435
Deploy a Message Consumer Plug-in	436
Enable the Message Consumer Plug-in for WS-Federation	436
Redirect Users After Failed Authentication Attempts	437
Supply SAML Attributes as HTTP Headers	438
Use Case for SAML Attributes As HTTP Headers	438
Configuration Overview to Supply Attributes as HTTP Headers	440
Set the Redirect Mode to Store SAML Attributes	440

Create an Authorization Rule to Validate Users.....	441
Configure a Response to Send Attributes as HTTP Headers	442
Create a Policy to Implement Attributes as HTTP Headers.....	443
How To Protect a Target Resource with a WS-Federation Authentication Scheme.....	443
Configure a Unique Realm for Each WS-Fed Authentication Scheme	444
Configure a Single Target Realm for All WS-Federation Authentication Schemes	445

Chapter 20: Use SAML 2.0 Provider Metadata To Simplify Configuration 449

SiteMinder SAML 2.0 Metadata Tools Overview	449
Export Metadata Tool	450
Run the smfedexport Tool	452
Command Options for smfedexport	453
smfedexport Tool Examples.....	455
Import Metadata Tool.....	457
Run the smfedimport Tool.....	458
smfedimport Tool Examples	459
Command Options for smfedimport.....	459
Processing Import Files with Multiple SAML 2.0 Providers.....	461
Processing Import Files with Multiple Certificate Aliases	461

Chapter 21: Federation Security Services Trace Logging 463

Trace Logging	463
FWS Log Messages at the Web Agent.....	463
Configure FWS Trace Logging.....	465
FWS Log Messages at the Policy Server	465
Use the SiteMinder Profiler to Log Trace Messages	466
Update Federation Web Services Data in the Logs	467
Simplify Logging with Trace Configuration Templates	467
Trace Logging Templates for FWS.....	467
Trace Logging Templates for the IdP and SP	469
Flush Federation Web Services Cache for Trace Logs	470

Chapter 22: Configuration Settings that Must Use the Same Values 471

How to Use the Configuration Settings Tables.....	471
SAML 1.x Matching Configuration Settings.....	471
SAML 2.0 Matching Configuration Settings.....	473
WS-Federation Configuration Settings.....	474

Chapter 23: Federation Web Services URLs Used by SiteMinder 477

Federation Services URLs	477
URLs for Services at the Asserting Party	477
Intersite Transfer Service URL (SAML 1.x)	478
Assertion Retrieval Service (SAML 1.x)	479
Artifact Resolution Service (SAML 2.0)	480
Single Sign On Service (SAML 2.0)	481
Single Sign-on Service (WS-Federation)	482
Single Logout Service at the IdP (SAML 2.0)	483
Signout Service at the AP (WS-Federation)	484
Identity Provider Discovery Profile Service (SAML 2.0)	485
Attribute Service (SAML 2.0)	485
WSFedDispatcher Service at the AP	487
URLs for Services at the Relying Party	487
SAML Credential Collector (SAML 1.x)	488
AuthnRequest (SAML 2.0)	489
Assertion Consumer Service (SAML 2.0)	490
Security Token Consumer Service (WS-Federation)	491
Single Logout Service at the SP (SAML 2.0)	492
Signout Service at the RP (WS-Federation)	493
WSFedDispatcher Service at the RP	494
The Web.xml File	494

Chapter 24: Troubleshooting 495

General Issues	495
Web Agent Option Pack Fails to Initialize Due to Invalid smjavaagent.dll	495
Cookie Domain Mismatch Errors	495
Error After Successful Authentication at Consumer/SP	496
HTTP 404 Error When Trying to Retrieve Assertion at the Consumer	496
Federation Web Services Fails to Send SAML Request to Producer/IdP	497
Matching Parameter Case-Sensitivity Configuration Issues	497
Error Message When Viewing FederationWSCustomUserStore	497
Policy Server System Fails After Logoff	498
Encrypted Private Key Fails to Be Imported into SMkeydatabase	498
Multibyte Characters in Assertions are Not Handled Properly	498
Trace Logs Not Appearing for IIS Web Server Using ServletExec	499
Error During Initialization of JVM	499
Affwebserver.log and FWSTrace.log Show Wrong Time	500
Resolving Signature Verification Failures	500
SAML 1.x-Only Issues	501
SAML 1.x Artifact Profile Single Sign-On Failing	501

Consumer Not Authenticating When Accessing Assertion Retrieval Service.....	502
Authentication Fails After Modifying Authentication Method	502
Client Authentication Fails for SAML Artifact Single Sign-on	502
SAML 2.0-Only Issues	503
SP Not Authenticating When Accessing Assertion Retrieval Service	503
ODBC Errors Deleting Expiry Data From Session Store	504

Appendix A: Federation Security Services Process Flow **505**

Flow Diagram for SSO Using SAML 1.x Artifact Authentication	505
Flow Diagram for SSO Using SAML 1.x POST Profile Authentication	508
Flow Diagram for SSO Using SAML 2.0 Authentication with Artifact Binding	510
Flow Diagram for SSO Using SAML 2.0 Authentication with POST Binding	514
Flow Diagram for WS-Federation SSO Initiated at the Resource Partner	518
WS-Federation SSO Initiated at the Account Partner	522
Flow Diagram for SAML 2.0 Single Logout	522
Flow Diagram for WS-Federation Signout (AP-initiated)	525
Flow Diagram for WS-Federation Signout (RP-initiated).....	528
Flow Diagram for Identity Provider Discovery Profile	530

Index **533**

Chapter 1: Federation Security Services Overview

This section contains the following topics:

[Introduction to SiteMinder Federation Security Services](#) (see page 19)

[Terminology for Partners in a Federation](#) (see page 20)

[SAML Profiles Supported by SiteMinder](#) (see page 20)

[WS-Federation](#) (see page 20)

[User Mapping](#) (see page 21)

[SiteMinder Components for Federation Security Services](#) (see page 22)

[Federated Single Sign-on with Security Zones](#) (see page 30)

[Federation Use Cases](#) (see page 31)

[SiteMinder Administrative User Interfaces](#) (see page 79)

Introduction to SiteMinder Federation Security Services

The growth of business networks provides opportunities for businesses to form partnerships to offer enhanced services to employees, customers, and suppliers. However, these new business opportunities present the following challenges:

- Exchanging user information between partners in a secure fashion
- Establishing a link between a user identity at a partner and a user identity in your company
- Enabling single sign-on across partner Web sites in multiple domains
- Handling different user session models between partner sites, such as single logout across all partner Web sites or separate sessions for each partner Web site
- Controlling access to resources based on user information received from a partner
- Interoperability across heterogeneous environments, such as Windows, UNIX operating systems and various Web servers, such as IIS, Sun Java System (formerly iPlanet/Sun ONE), and Apache

SiteMinder Federation Security Services provides a solution to all these challenges.

Note: Federation Security Services is separately-licensed from SiteMinder.

Terminology for Partners in a Federation

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

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

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

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

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

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

SAML Profiles Supported by SiteMinder

Federation Security Services supports the following SAML standards and profiles:

- SAML 1.0 Artifact profile only
- SAML 1.1 Artifact and POST profile
- SAML 2.0 Artifact and POST profile

WS-Federation

Active Directory Federation Services (ADFS) is web services-based solution from Microsoft for federated single sign-on (SSO). ADFS runs on a Windows server and accomplishes SSO by letting partners securely share user identity information and access rights across a secure network. ADFS extends SSO functionality to internet applications, letting users have a seamless web SSO interaction when they access web-based applications of the organization.

ADFS uses the following specifications:

- Web Services Federation (WS-Federation)
- WS-Federation Passive Requestor Profile (WS-F PRP)
- WS-Federation Passive Requestor Interoperability Profile

For WS specifications and background documentation, and information about ADFS profiles, go to the [Microsoft website](#).

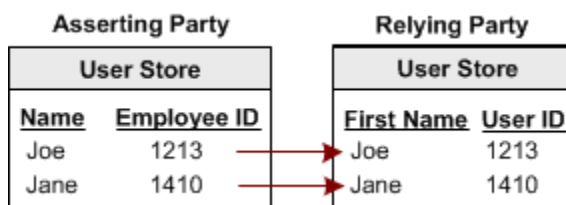
User Mapping

User mapping is the ability to establish a relationship between a user identity at one business and a user identity at another business. This relationship is established by mapping remote users at an asserting party to local users at the relying party.

The types of mapping are as follows:

- One-to-one mapping maps a unique remote user directory entry at the asserting party to a unique user entry at the relying party.

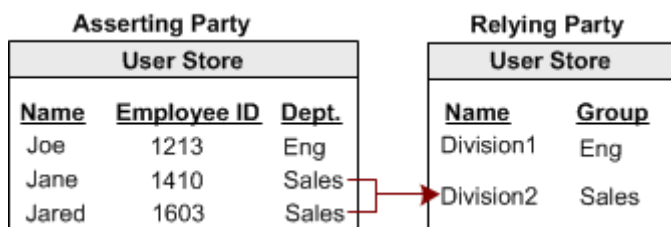
One-to-one mapping is often referred to as account linking, as it links an account at an asserting party site to an account at a relying party, as shown in the following illustration:



- N-to-one mapping maps a group of remote user directory entries to a single local profile entry.

N-to-one mapping allows several user records at a asserting party to be mapped to one user record or profile at the relying party. An administrator at the relying party can use this type of mapping to define access control for a group of remote users, without having to maintain a record for each remote user.

The following illustration shows n-to-one mapping:



SiteMinder Components for Federation Security Services

SiteMinder's Federation Security Services solution encompasses several components:

- SAML Assertion Generator—A Policy Server component that creates SAML assertions at a producer site.
- WS-Federation Assertion Generator—A Policy Server component that creates WS-Federation RequestSecurityTokenResponse messages containing SAML assertions.
- SAML and WS-Federation Authentication Schemes—A Policy Server component that validates SAML or WS-Federation assertions and maps assertion data to a local user at a site that consumes assertions. The supported authentication schemes are: SAML 1.x artifact, SAML 1.x POST, and SAML 2.0 (artifact and POST binding), and WS-Federation.
- Federation Web Services—A Web Agent component that supports assertion retrieval, session synchronization and notification alerts at an asserting party site, as well as collecting assertions at a relying party.
- SAML Affiliate Agent—A stand-alone component that provides authentication and session management capabilities to a consumer site that does not use a SiteMinder Policy Server and Web Agent. This Agent only supports SAML 1.0.

Note: When the SAML Affiliate Agent is the consumer, the Web Agent provides access to the SAML assertion generator.

SAML Assertion Generator

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

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

- SAML artifact profile/binding
The assertion generator stores the assertion in the SiteMinder session server. A reference to the assertion is returned to the Web Agent in the form of a SAML artifact.
- SAML POST profile/binding
SiteMinder returns the assertion by way of a browser as a SAML response embedded in an HTTP form.

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

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

WS-Federation Assertion Generator

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

The Web Agent is responsible for sending the WS-Federation security token response message, via a user's browser, to the site that consumes the assertion in accordance with the WS-Federation Passive Requestor profile. At the Resource Partner, a client, such as WS-Federation Assertion Consumer must be available to process the assertion.

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

The assertion generator is installed by the Policy Server. After installing the Policy Server, the Account Partner administrator can use the FSS Administrative UI to define and configure affiliates.

SAML and WS-Federation Authentication Schemes

SiteMinder supports the following authentication schemes:

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

Each authentication scheme enables a SiteMinder site to consume SAML assertions. Upon receiving an assertion, the authentication scheme validates the SAML assertion, maps assertion data to a local user, and establishes a SiteMinder session at the site consuming the assertion.

One of the critical features of the SAML authentication schemes is to map remote users at an asserting party to local users at the relying party. The mapping is defined as part of the authentication scheme configuration. User mapping information enables the authentication scheme to locate the correct user record for authentication.

The SAML and WS-Federation authentication schemes are installed by the Policy Server. After installation, the administrator can use the FSS Administrative UI to define and configure these schemes and use them to protect specific resources.

Customizing SAML 2.0 Assertion Responses

You can implement your own business logic in addition to the standard SAML authentication processing using the Message Consumer Plug-in. This plug-in lets you further manipulate a SAML 2.0 assertion response, which is part of the SAML 2.0 authentication processing.

The Message Consumer Plug-in is SiteMinder's Java program that implements the SAML 2.0 Message Consumer Extension API. The plug-in can be integrated using settings provided by the SAML 2.0 authentication scheme.

Federation Web Services Application

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

SAML 1.x Artifact and POST Profiles

For the SAML 1.x artifact and POST profiles, the Federation Web Services application uses the following services:

Assertion Retrieval Service (SAML 1.x Artifact only)

A producer-side component. This service handles a SAML request for the assertion that corresponds to a SAML artifact by retrieving the assertion from the SiteMinder session store. The SAML specification defines the assertion retrieval request and response behavior.

Note: Only the SAML artifact profile uses the assertion retrieval service.

Session Synchronization (SAML 1.x)

A producer-side component that validates and terminates sessions for the SAML Affiliate Agent (A SiteMinder value-added service, which uses a standards-based SOAP RPC mechanism).

Notification Alert (SAML 1.x)

A producer-side component that logs resource access notification events for the SAML Affiliate Agent (A SiteMinder value-added service, which uses a standards-based SOAP RPC mechanism).

SAML Credential Collector (SAML 1.x)

A consumer-side component that receives a SAML artifact or an HTTP form with an embedded SAML response and obtains the corresponding SAML assertion. The credential collector issues SiteMinder cookies to a browser of the user.

Intersite Transfer Service (SAML 1.x)

A producer-side component for the SAML POST profile. The intersite transfer service transfers a user from the producer site to a consumer site. For the SAML artifact profile, the Web Agent performs the same function as the intersite transfer service.

SAML 2.0 Artifact and POST Profiles

For SAML 2.0 artifact and POST profiles, the Federation Web Services application uses the following services:

Artifact Resolution Service (SAML 2.0 Artifact only)

An Identity Provider-side service that corresponds to the SAML 2.0 authentication using the HTTP-artifact binding. This service retrieves the assertion stored in the SiteMinder session store at the Identity Provider.

Note: Only the HTTP-artifact binding uses the artifact resolution service.

Assertion Consumer Service (SAML 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 SiteMinder cookies to a browser.

Note: The Assertion Consumer Service accepts an AuthnRequest with an AssertionConsumerServiceIndex value of 0. All other values for this setting are denied.

AuthnRequest Service (SAML 2.0)

This service is deployed for use by SAML 2.0. A Service Provider can generate an <AuthnRequest> message to authenticate a user for cross-domain single sign-on. This message contains information that enables the Federation Web Services application to redirect the browser to the single sign-on service at the Identity Provider. The AuthnRequest service is used for POST and Artifact single sign-on.

Single Sign-on Service (SAML 2.0)

The single sign-on service enables an Identity Provider to process AuthnRequest messages. The service also invokes the assertion generator to create an assertion that is sent to the Service Provider.

Single Logout Service (SAML 2.0)

This service implements processing of single logout functionality, which an Identity Provider or a Service Provider can initiate.

Identity Provider Discovery Service (SAML 2.0)

Implements SAML 2.0 Identity Provider Discovery Profile and sets and retrieves the common domain cookie. An IdP requests to set the common domain cookie after authenticating a principal. An SP requests to obtain the common domain cookie to discover which Identity Provider a principal is using.

WS-Federation Passive Requestor Profile

For the WS-Federation Passive Requestor profile, the Federation Web Services application uses the following services:

Note: WS-Federation is only available with Federation Security Services.

Security Token Consumer Service

A Resource Partner component that receives a security token and extracts the corresponding SAML assertion. The Security Token Consumer Service issues SiteMinder cookies to a browser.

Single Sign-on Service

Enables an Account Partner to process a wsignin message and gather the necessary Resource Partner information to authenticate the user. This service also invokes the assertion generator to create an assertion that is sent to the Resource Partner.

Signout Service

Implements processing of single logout functionality by way of a signout servlet. An Account Partner or a Resource Partner can initiate signout.

SAML Affiliate Agent

The SAML Affiliate Agent enables businesses using the Policy Server and Web Agent to act as a main portal and share security and customer profile information with affiliated partners. The affiliated partners use only the SAML Affiliate Agent.

Note: The SAML Affiliate Agent only supports SAML 1.0 and it is not FIPS-compatible.

The SAML Affiliate Agent is a stand-alone component. This agent provides single sign-on and session management capabilities to a third-party consumer. The consumer, or affiliate, does not maintain identities for users at the producer, or portal, site. The affiliate site can determine that the user has been registered at the portal site, and optionally, that the user has an active SiteMinder session at the portal site. Based on the affiliate policies at the portal, information can be passed to the affiliate and set as cookies or header variables for the affiliate web server.

For more information about the SAML Affiliate Agent, see the *SiteMinder SAML Affiliate Agent Guide*.

Secure Proxy Server Federation Gateway

The SiteMinder Secure Proxy Server (SPS) federation gateway offers a proxy-based solution to access control in a federated network. Unlike a traditional proxy, which typically serves a group of users requesting Internet resources, the SPS federation gateway is a reverse proxy, meaning it acts on behalf of users requesting resources from an enterprise.

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

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

Note: The Secure Proxy Server is an separately-licensed product from SiteMinder.

Debugging Features

The Federation Security Services components log specific events to monitor and debug activity across the federated network.

- Web Agent log—Displays information about any request to generate a SAML assertion at a producer site
- Federation Web Services log—Displays information about requests to retrieve SAML assertions and to consume SAML assertions. Additionally, the log displays session synchronization and notification events from the SAML Affiliate Agent.
- Policy Server log—Displays the results of calls from the SAML assertion generator and SAML artifact authentication scheme. Also displays Policy Server trace messages that you configure using the Policy Server Management Profiler or using one of the provided profiler template files.
- Web Agent Option Pack logs—Displays FWS trace messages that you configure using the FWSTrace.conf file or using one of the provided trace template files.
- SAML Affiliate Agent logs—Displays information about activities at a consumer site protected by the SAML Affiliate Agent.

APIs for Federation Security Services

The following APIs provide support for Federation Security Services.

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

Policy Management API

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

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

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

Java Message Consumer Plugin API

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

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

Java Assertion Generator Plugin API

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

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

Internationalization in Federation Security Services

Federation Security Services supports the following features for I18N internationalization:

- Federation Security Services configuration objects, Java and C++ code are encoded in UTF-8 format for the internationalization purposes.
- SiteMinder supports the creation and consumption of default and customized SAML 1.x, SAML 2.0, and WS-Federation assertions with multibyte user ids and attribute values.
- All target and redirect URLs are encoded per HTTP 1.1 RFC 2616 so multibyte path and file names are handled correctly.

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

```
LANG=xx_xx.UTF-8
```

For example, for Japanese, the entry would be:

```
LANG=ja_JP.UTF-8
```

Federated Single Sign-on with Security Zones

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

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

These features include:

Artifact Single sign-on

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

Single Logout

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

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

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

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

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

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

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

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

SSOZoneName

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

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

SSOTrustedZone

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

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

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

Federation Use Cases

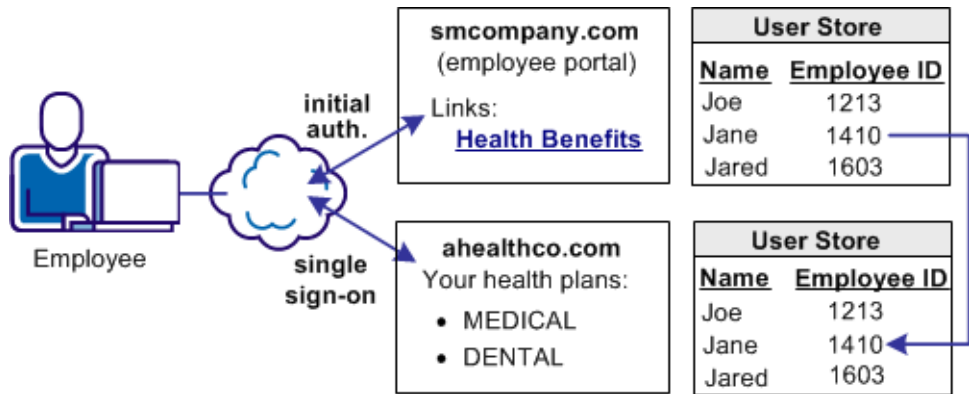
There are probably as many use cases for federated networks as there are business arrangements between partners. This section presents use cases that demonstrate different ways of handling user identities to provide single sign-on and single logout between partners.

Use Case 1: Single Sign-on Based on Account Linking

In Use Case 1, smcompany.com contracts with a partner company, ahealthco.com to manage employee health benefits.

An employee of smcompany.com authenticates at an employee portal at his company's site, www.smcompany.com and clicks a link to view her health benefits at ahealthco.com. The employee is taken to the ahealthco.com web site and is presented with her health benefit information without having to sign on to the website.

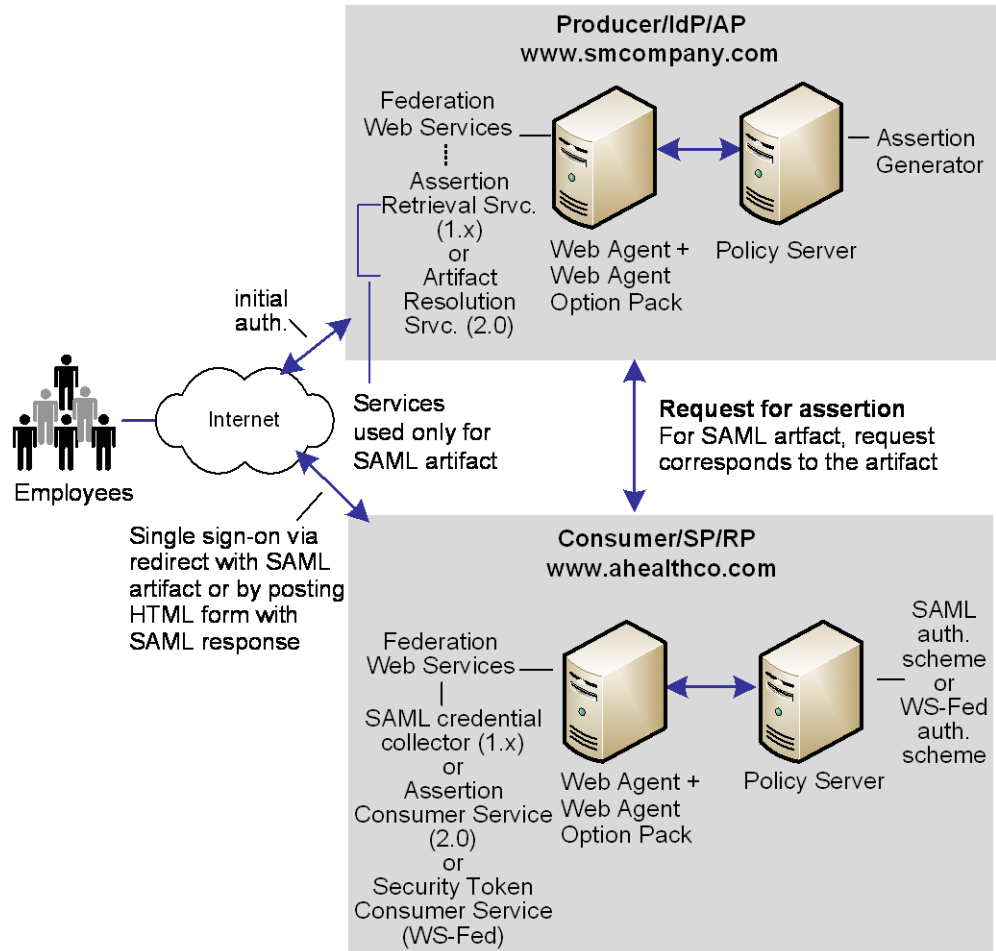
The following illustration shows this use case.



The company, ahealthco.com, maintains all health-related information for employees at smcompany.com. To do this, ahealthco.com maintains user identities for every employee of smcompany.com. When an employee of smcompany.com accesses ahealthco.com, an identifier for the employee is passed from smcompany.com to ahealthco.com in a secure manner. This identifier allows ahealthco.com to determine who the user is and the level of access to allow for that user.

Solution 1: Single Sign-on based on Account Linking

Solution 1 illustrates how Federation Security Services can be deployed at smcompany.com and ahealthco.com to solve [Use Case 1: Single Sign-on Based on Account Linking](#) (see page 32).



SiteMinder is deployed at both sites. The Web Agent with the Web Agent Option Pack are installed on a webserver system and the Policy Server is installed on another system. The installations are the same for smcompany.com and ahealthco.com.

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

Solution 1 Using SAML 1.x Artifact Authentication

In this example, smcompany.com is acting as the producer site. When an employee of smcompany.com accesses an employee portal at www.smcompany.com, the sequence of events is as follows:

1. The Web Agent provides the initial authentication.
2. When the employee clicks a link at smcompany.com to view the health benefits at ahealthco.com, the link makes a request to the Intersite Transfer Service at www.smcompany.com.
3. The Intersite Transfer Service calls the assertion generator, which creates a SAML assertion, puts the assertion into the SiteMinder session store. The service returns a SAML artifact.
4. The Web Agent redirects the user to www.ahealthco.com with the SAML artifact, in accordance with the SAML browser artifact protocol.

Ahealthco.com is acting as the consumer site. The SAML credential collector service handles the redirect request of the SAML artifact. The credential collector is part of the Federation Web Services application at ahealthco.com.

The sequence of events is as follows:

1. The SAML credential collector calls the SAML artifact authentication scheme to obtain the location of the assertion retrieval service at smcompany.com.
2. The SAML credential collector calls the assertion retrieval service at www.smcompany.com.
3. The assertion retrieval service at www.smcompany.com retrieves the assertion from the SiteMinder session store and returns it to the SAML credential collector at ahealthco.com.
4. The SAML credential collector then passes the assertion to the SAML artifact authentication scheme for validation and session creation. Also, it issues a SiteMinder session cookie to the browser.
5. The user is allowed access to resources at ahealthco.com based on the policies that are defined at the Policy Server at ahealthco.com. The Web Agent at ahealthco.com enforces the policies.

In this example, the administrator at smcompany.com configures an affiliate for ahealthco.com. The affiliate is configured with an attribute that is a unique ID for the user. This action causes the assertion generator to include that attribute as part of the user profile in a SAML assertion that is created for ahealthco.com.

The administrator at ahealthco.com configures a SAML artifact authentication scheme for smcompany.com. The authentication scheme specifies the location of the assertion retriever service at smcompany.com. The scheme also extracts the unique user ID from the SAML assertion, determines how to search the ahealthco.com user directory for the user record that matches the value from the assertion.

Solution 1 Using SAML 1.x POST Profile

In this example, smcompany.com is acting as the producer site. When an employee of smcompany.com accesses an employee portal at www.smcompany.com, the sequence of events is as follows:

1. The Web Agent provides the initial authentication.
2. When the employee clicks a link at www.smcompany.com to view the health benefits at ahealthco.com, the link makes a request to the Intersite Transfer Service at www.smcompany.com.
3. The Intersite Transfer Service calls the assertion generator, which creates a SAML assertion and signs the SAML response.
4. The signed response is then placed in an auto-POST HTML form and sent to the browser of the user.
5. The browser automatically POSTs a form to the Assertion Consumer URL (which is the SAML credential collector), at ahealthco.com. The form contains a SAML response as a form variable.

Ahealthco.com is acting as the consumer site. The SAML credential collector service handles the redirect request with the SAML response. The SAML credential collector that is part of the Federation Web Services at ahealthco.com.

The sequence of events is as follows:

1. The SAML credential collector calls for the requested target resource at ahealthco.com. The SAML POST profile authentication scheme protects the target resource.
2. Because the SAML POST profile scheme is protecting the resource, the SAML credential collector decodes the SAML response message.
3. Using the digitally signed SAML response message as credentials, the SAML credential collector calls the Policy Server at ahealthco.com.
4. The Policy Server verifies the signature and then authenticates the user using the SAML assertion embedded in the decoded SAML response message. Based on the assertion, the Policy Server lets the user log in.

5. After the user logs in, the SAML credential collector creates an SMSESSION cookie, places it in the browser of the user, and redirects the user to the target resource at ahealthco.com.
6. The user is allowed access to resources at ahealthco.com based on policies that are defined at the Policy Serve. The Web Agent enforces the policies.

In this example, the administrator at smcompany.com uses the UI to configure an affiliate object for ahealthco.com. The affiliate is configured with an attribute that is a unique ID for the user. This action causes the assertion generator to include that attribute as part of the user profile in a SAML assertion that is created for ahealthco.com.

The administrator at ahealthco.com configures a SAML POST profile authentication scheme for smcompany.com. The authentication scheme specifies how to extract the unique user ID from the SAML assertion. The scheme also defines how to search the user directory at ahealthco.com for the user record that matches the value from the assertion.

Solution 1 Using SAML 2.0 Artifact Authentication

In this example, smcompany.com is acting as the Identity Provider. When an employee of smcompany.com accesses an employee portal at www.smcompany.com, the sequence of events is as follows:

1. The Web Agent provides the initial authentication. When the user clicks a link at the Identity Provider, this action is referred to as an unsolicited response at the Identity Provider.
2. When the employee clicks a link at www.smcompany.com to view the health benefits at ahealthco.com, the link makes a request to the Single Sign-on Service at www.smcompany.com.
3. The single sign-on service calls the assertion generator creates a SAML assertion, stores the assertion in the SiteMinder session store, and returns a SAML artifact.
4. The Web Agent redirects the user to ahealthco.com with the SAML artifact, in accordance with the SAML browser artifact protocol.

Ahealthco.com is acting as the Service Provider. One of the components of the Service provider is the Assertion Consumer Service. The Assertion Consumer Service handles the redirect request containing the SAML artifact.

The sequence of events is as follows:

1. The Assertion Consumer Service calls the SAML 2.0 authentication scheme with HTTP-artifact binding to obtain the location of the artifact resolution service at smcompany.com.
2. The Assertion Consumer Service calls the artifact resolution service at www.smcompany.com.

3. The artifact resolution service at www.smcompany.com retrieves the assertion from the session store at smcompany.com and returns it to the artifact resolution service at ahealthco.com.
4. The Assertion Consumer Service passes the assertion to the SAML 2.0 authentication scheme for validation and session creation. The service issues a SiteMinder session cookie to the browser.
5. The user is allowed access to resources at ahealthco.com based on policies that are defined at the Policy Server at ahealthco.com. The Web Agent at ahealthco.com enforces the policies.

In this example, the administrator at smcompany.com configures a Service Provider object for ahealthco.com. The Service Provider is configured with an attribute that is a unique ID for the user. This action causes the assertion generator to include that attribute as part of the user profile in an assertion that is created for ahealthco.com.

The administrator at ahealthco.com configures a SAML 2.0 authentication scheme that uses the artifact binding for smcompany.com. The authentication scheme has the following information:

- The location of the artifact resolution service at smcompany.com.
- How to extract the unique user ID from the SAML assertion.
- How to search the user directory at ahealthco.com for the user record that matches the value in the assertion.

Solution 1 Using SAML 2.0 POST Binding

In this example, smcompany.com is acting as the Identity Provider. When an employee of smcompany.com accesses an employee portal at www.smcompany.com, the sequence of events is as follows:

1. The Web Agent provides the initial authentication. When the user clicks a link at the Identity Provider, this action is referred to as an unsolicited response at the Identity Provider.
2. When the employee clicks a link at www.smcompany.com to view the health benefits at ahealthco.com, the link makes a request to the Single Sign-on Service at www.smcompany.com.
3. The Single Sign-on Service passes calls the assertion generator, which creates a SAML assertion and signs the SAML response.
4. The signed response is then placed in an auto-POST HTML form and sent to the browser of the user.
5. The browser automatically POSTs a form to the Assertion Consumer URL at ahealthco.com. The form contains a SAML response as a form variable.

Ahealthco.com is acting as the Service Provider. The Assertion Consumer Service handles the redirect request with the SAML response. The service is part of the Federation Web Services at ahealthco.com.

The sequence of events is as follows:

1. The Assertion Consumer Service calls for the requested target resource at ahealthco.com. The SAML 2.0 authentication scheme protects this resource using the HTTP-POST binding.
2. Because the SAML 2.0 authentication scheme is protecting the resource, the Assertion Consumer Service passes the digitally signed SAML response message as credentials, to the Policy Server at ahealthco.com.
3. The Policy Server verifies the signature and then authenticates the user using the SAML assertion embedded in the decoded SAML response message. Based on the assertion, the Policy Server lets the user log in.
4. After the user logs in, the Assertion Consumer Service creates an SMSESSION cookie, places it in the browser of the user, and redirects the user to the target resource at ahealthco.com.
5. The user is allowed access to resources at ahealthco.com based on policies that are defined at the Policy Server. The Web Agent enforces the policies.

In this example, the administrator at smcompany.com configures a Service Provider object for ahealthco.com. The Service Provider is configured with an attribute that is a unique ID for the user. This action causes the assertion generator to include that attribute as part of the user profile in a SAML assertion that is created for ahealthco.com.

The administrator at ahealthco.com configures a SAML 2.0 authentication scheme with the HTTP-POST binding for smcompany.com. The authentication scheme has the following information:

- How to extract the unique user ID from the SAML assertion.
- How to search the user directory at ahealthco.com for the user record that matches the value from the assertion.

Solution 1 Using WS-Federation Passive Requestor Profile

In this example, smcompany.com is acting as the Account Partner. When an employee of smcompany.com accesses an employee portal at www.smcompany.com, the sequence of events is as follows:

1. The user visits an unprotected site selection page at ahealthco.com.
2. This link points to the Single Sign-on Service at the Account Partner, www.smcompany.com. The Web Agent provides the initial authentication.

3. The Single Sign-on Service calls the WS-Federation Assertion Generator, which creates a SAML 1.1 assertion. WS-Federation Assertion Generator signs the assertion and wraps the assertion in a security token response message.
4. The response is then placed in an auto-POST HTML form as a form variable and sent to the browser of the user.
5. The browser automatically POSTs a form to the Security Token Consumer Service URL at ahealthco.com.

Ahealthco.com is acting as the Resource Partner. The Security Token Consumer Service handles the redirect request with the SAML response. The service is part of the Federation Web Services application.

The sequence of events is as follows:

1. The Security Token Consumer Service calls for the requested target resource at ahealthco.com. The WS-Federation authentication scheme protects this resource.
2. Because the WS-Federation authentication scheme is protecting the resource, the Security Token Consumer Service passes the signed assertion as credentials to the Policy Server at ahealthco.com.
3. The Policy Server verifies the signature and then authenticates the user using the SAML assertion embedded in the decoded SAML response message. Based on the assertion, the Policy Server lets the user log in.
4. After the user logs in, the Security Token Consumer Service creates an SMSESSION cookie. The service then places the cookie in the browser and redirects the user to the target resource at ahealthco.com.
5. The user is allowed access to resources at ahealthco.com based on policies that are defined at the Policy Server. The Web Agent enforces the policies.

In this example, the administrator at smcompany.com configures a Resource Partner object for ahealthco.com. The Resource Partner is configured to include an attribute in the assertion that is a unique ID for the user. The assertion generator includes that attribute in the SAML assertion for ahealthco.com.

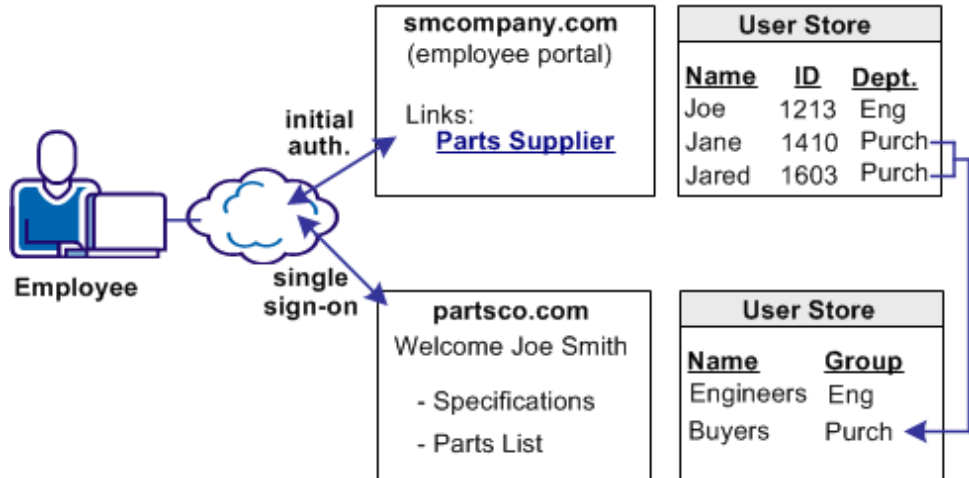
The administrator at ahealthco.com configures a WS-Federation authentication scheme for smcompany.com. The authentication scheme has the following information:

- How to extract the unique user ID from the SAML assertion.
- How to search the user directory at ahealthco.com for the user record that matches the value from the assertion.

Use Case 2: Single Sign-on Based on User Attribute Profiles

In Use Case 2, smcompany.com buys parts from a partner named partsco.com.

An engineer authenticates at the employee portal, smcompany.com and clicks a link to access information at partsco.com. Being an engineer at smcompany.com, the user is taken directly to the Specifications portion of the partsco.com website without having to log in.



When a buyer for smcompany.com authenticates and clicks a link for partsco.com, the buyer is taken directly to the Parts List portion of the partsco.com website. The buyer does not have to log in.

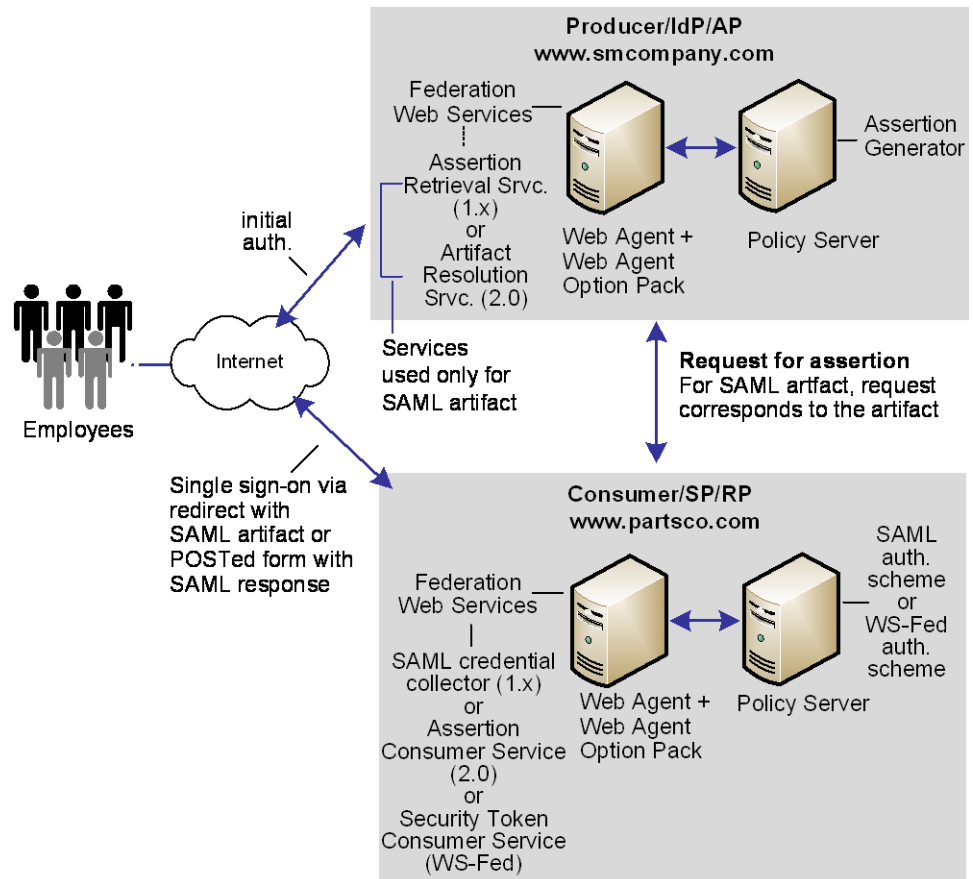
Additional attributes, such as the user name are passed from smcompany.com to partsco.com to personalize the interface for the individual user.

Partsco.com does not want to maintain user identities for all employees at smcompany.com, but the company wants to control access to sensitive portions of the website. To control the access, partsco.com maintains a limited number of profile identities for users at smcompany.com. One profile identity is maintained for engineers and one profile identity is maintained for buyers.

When an employee of smcompany.com accesses partsco.com, smcompany.com sends user attributes in a secure manner to partsco.com. Partsco.com uses the attributes to determine what profile identity controls access.

Solution 2: Single Sign-on based on User Attribute Profiles

Solution 2 shows how Federation Security Services can be deployed at smcompany.com and partsco.com to solve [Use Case 2: Single Sign-on Based on User Attribute Profiles](#) (see page 40).



SiteMinder is deployed at both sites. The interactions between the user and each site is similar, where partsco.com is acting as the relying party.

The following illustration is similar for SAML 1.x, SAML 2.0, and WS-Federation; however, the Federation Web Services components are different as follows:

- For SAML 1.x, the Artifact Resolution Service (artifact profile only) is at the IdP and the SAML credential collector is at the SP.
- For SAML 2.0, the Assertion Retrieval Service (artifact binding only) is at the IdP and the Assertion Consumer Service at the SP.
- For WS-Federation, the Single Sign-on Service is at the IdP and the Security Token Consumer Service is at the SP.

Note: WS-Federation only supports HTTP-POST binding.

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

The configuration is similar to Solution 1: Single Sign-on based on Account Linking, except for the following items:

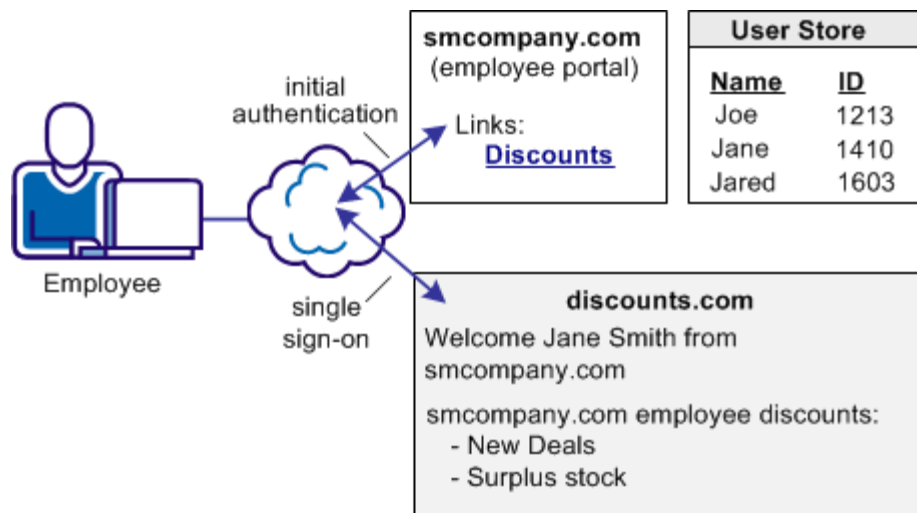
- The administrator at smcompany.com defines the consumer/SP for partsco.com with an attribute specifying the department of the user at the company. The assertion generator includes this attribute as part of the user profile in the assertion it creates for partsco.com.
- The administrator at partsco.com defines an authentication scheme (artifact, post, or WS-federation) for smcompany.com. The scheme extracts the department attribute from the SAML assertion. The scheme then searches the user directory at partsco.com for the user record that matches the department value from the assertion. The administrator defines one user profile record for each department that is allowed to access the partsco.com website.

Use Case 3: Single Sign-on with No Local User Account

In Use Case 3, smcompany.com offers employee discounts by establishing a partnership with discounts.com.

An employee of smcompany.com authenticates at smcompany.com and clicks a link to access discounts.com. The employee is taken to the discounts.com website and presented with the discounts available for smcompany.com employees, without logging in to the discounts.com website.

The following illustration shows this use case.



Discounts.com does not maintain any identities for smcompany.com. The company allows all employees of smcompany.com to access discounts.com as long as long as they have been authenticated at smcompany.com. When an employee of smcompany.com accesses discounts.com, authentication information is sent in a secure manner from smcompany.com to discounts.com. This information is used to allow access to discounts.com.

Additional attributes, such as the user name are passed from smcompany.com to discounts.com to personalize the interface for the individual user.

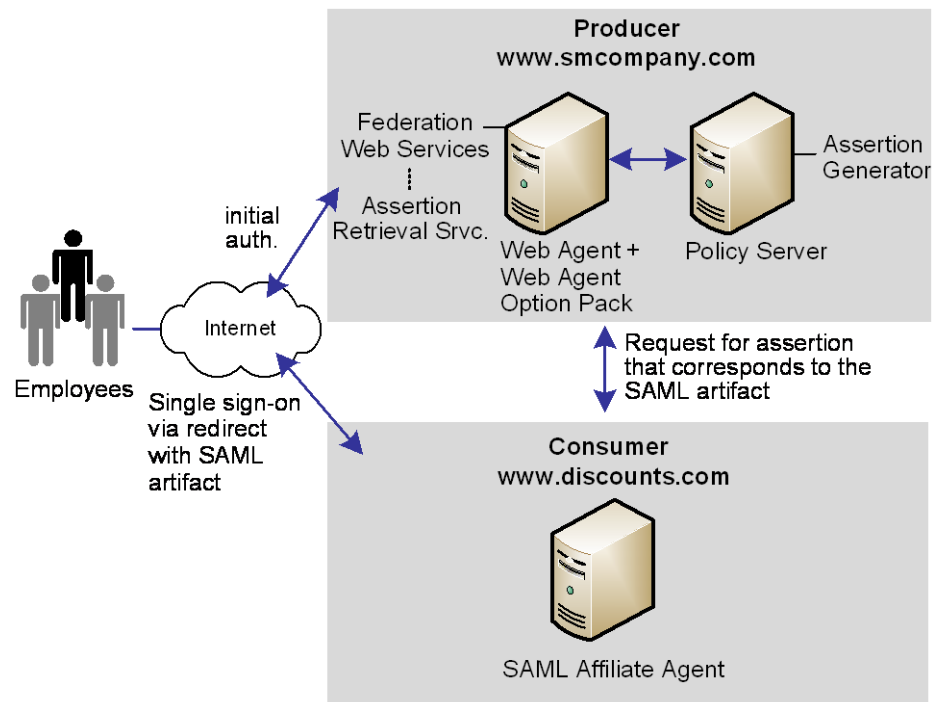
Solution 3: Single Sign-on with no Local User Account

Solution 3 shows how SiteMinder Federation Security Services can be deployed at smcompany.com and discounts.com to solve [Use Case 3: Single Sign-on with No Local User Account](#) (see page 42).

SiteMinder is deployed at smcompany.com by installing the Web Agent with the Web Agent Option pack on one system, and installing the Policy Server on another system. The SAML Affiliate Agent is installed at discounts.com.

Note: The SAML Affiliate Agent only supports SAML 1.0 and is not FIPS-compatible.

The following figure shows single sign-on with no local user account.



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

Smcompany.com is acting as a SAML 1.x producer. When an employee of smcompany.com accesses an employee portal at www.smcompany.com, the following process occurs:

1. The Web Agent provides the initial authentication.
2. When the employee clicks a link at www.smcompany.com to access deals at discounts.com, the link makes a request to the Web Agent at www.smcompany.com.
3. The Web Agent at www.smcompany.com calls the assertion generator. The assertion generator creates a SAML assertion and stores the assertion in the SiteMinder session store. Finally, smcompany.com returns a SAML artifact to discounts.com.
4. The Web Agent redirects the user to www.discounts.com with the SAML artifact in accordance with the SAML browser artifact protocol.

Discounts.com is acting as the consumer site. The SAML Affiliate Agent at www.discounts.com handles the redirect request with the SAML artifact, as follows:

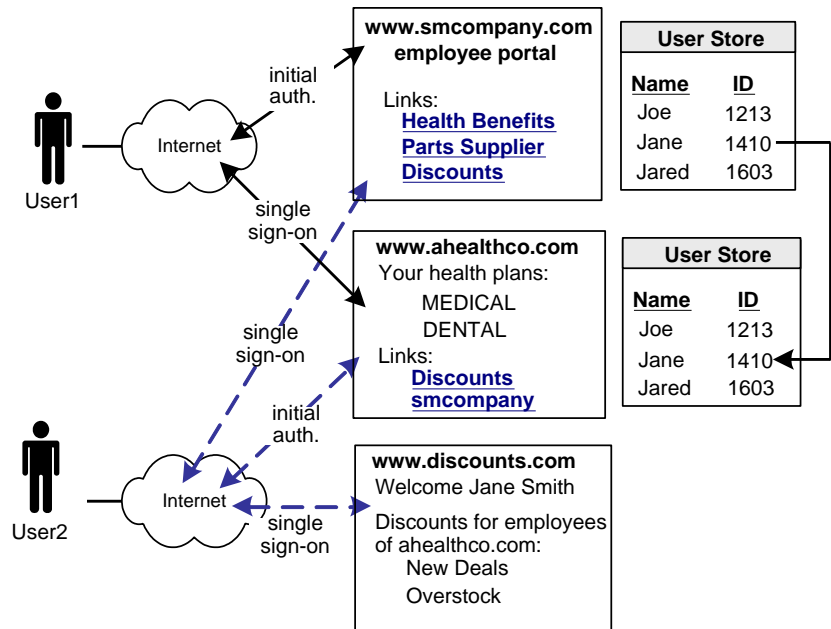
1. The SAML Affiliate Agent obtains the location of the assertion retrieval service at www.smcompany.com from a configuration file.
2. The SAML Affiliate Agent calls the assertion retrieval service at www.smcompany.com.
3. The assertion retrieval service at www.smcompany.com retrieves the assertion from the SiteMinder session store and returns it to the SAML affiliate agent at www.discounts.com.
4. The SAML Affiliate Agent then validates the SAML assertion and issues a SiteMinder affiliate session cookie to the browser.
5. The user is allowed access to resources at discounts.com.

The administrator at smcompany.com uses the Policy Server User Interface to configure an affiliate for discounts.com. The affiliate is configured to include attributes in the assertion. The assertion generator includes the attributes in the SAML assertion it creates for discounts.com.

The administrator at discounts.com configures the SAML Affiliate Agent with information about the discounts.com site, the location of the assertion retriever service at smcompany.com, and the resources the affiliate protects.

Use Case 4: Extended Networks

In Use Case 4, smcompany.com, ahealthco.com, and discounts.com all participate in an extended federated network. This case is an extension of the previous use cases.



In this network, not all customers of ahealthco.com work at smcompany.com. Ahealthco.com provides discounts only to its customers by establishing a relationship between themselves and discounts.com. Ahealthco.com maintains user identities for every customer so ahealthco.com manages local credentials, such as a password for each user. By managing local credentials, ahealthco.com can authenticate users and can provide single sign-on access to its partners.

In this extended network, the users access each website differently:

- User1 accesses health benefit information at the ahealthco.com website. User1 can access the partsco.com website by clicking the PartsSupplier link at smcompany.com, the employee portal. User1 can also click a link at the employee portal to access discounts at discounts.com.

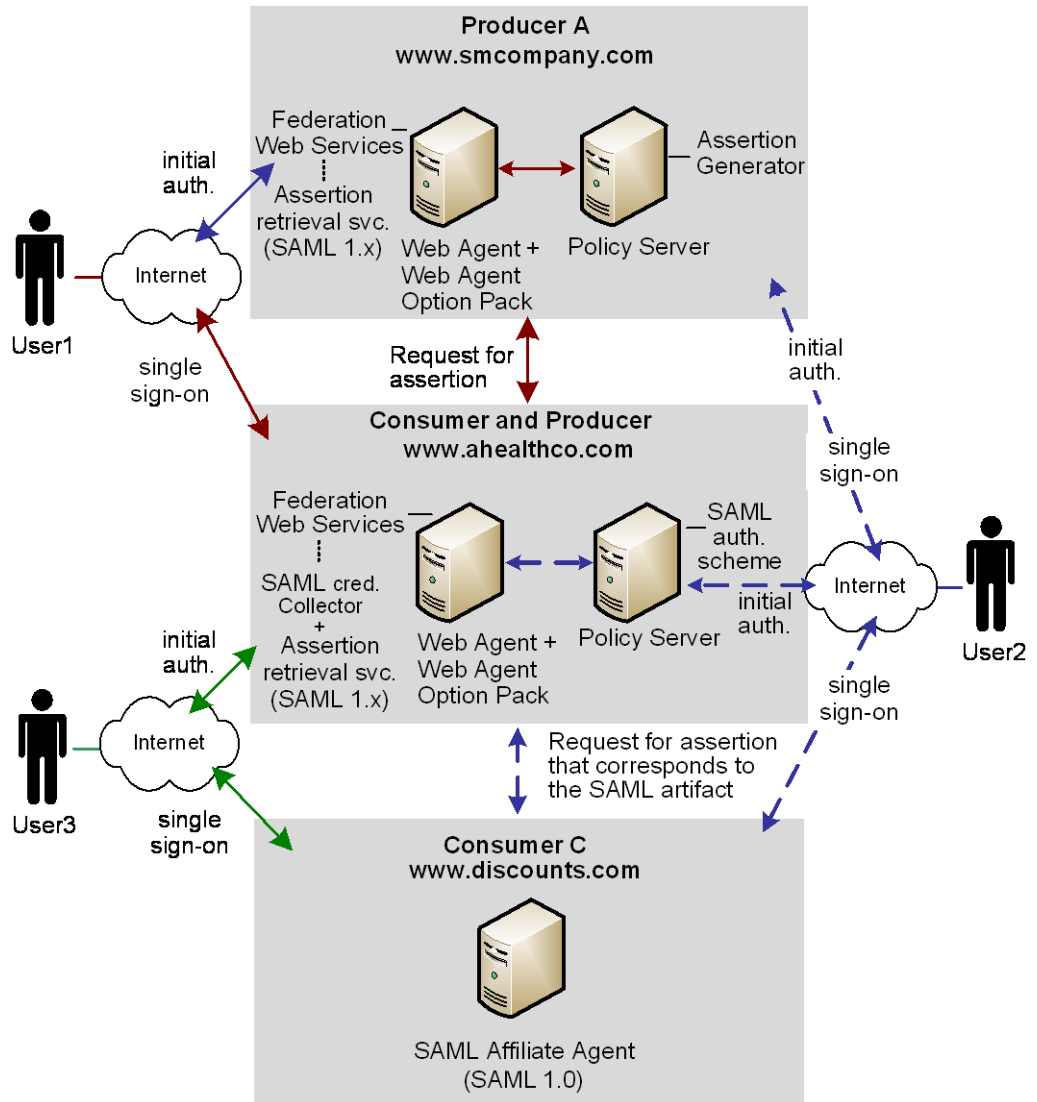
User2 authenticates at the ahealthco.com website and clicks a link to access discounts at discounts.com, without logging in to the discounts.com website. The discounts the site presents to User2 reflect the business arrangement between ahealthco.com and discounts.com. Being employee of smcompany.com, User2 can also click a link at ahealthco.com and access the employee portal at smcompany.com without logging in to website.

- User3 (not shown in the example), is a customer of ahealthco.com, but is not an employee of smcompany.com. User3 authenticates at the ahealthco.com website and clicks a link to access discounts at discounts.com. User3 does not log in to the website. The discounts the site presents to User3 reflect the business arrangement between ahealthco.com and discounts.com. Because User3 is not an employee of smcompany.com, User3 cannot access the smcompany.com website.

Solution 4: Extended Networks

Solution 4 illustrates how Federation Security Services can be deployed at smcompany.com, ahealthco.com, and discounts.com to solve [Use Case 4: Extended Networks](#) (see page 45).

The following illustration shows an extended network. SAML 1.x is the protocol in use.



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

SiteMinder is deployed at smcompany.com and ahealthco.com. At each site, the Web Agent and Web Agent Option Pack is installed on one system and the Policy Server on another system. The SAML Affiliate Agent is installed at discounts.com.

In Solution 4:

- smcompany.com acts as a producer for User1 and a consumer for User2.
- ahealthco.com acts as a consumer for User1 and a producer for User2 and a producer for User3.
- discounts.com acts as a consumer for User1, User2, and User3.

The administrator for smcompany.com has configured two entities in an affiliate domain, which represents ahealthco.com and discounts.com. These sites are configured in a similar manner as in Examples 1 and 3 described previously, but the configurations have been extended as follows:

- At smcompany.com, the administrator has configured a SAML authentication scheme (artifact or POST). For User2, the authentication scheme enables smcompany.com to act as a consumer for ahealthco.com.
- At ahealthco.com:
 - The administrator has configured an affiliate object that represents smcompany.com so an assertion is produced for User2. This configuration makes single sign-on to smcompany.com possible.
 - The administrator has configured an affiliate object that represents discounts.com so an assertion is produced for User2 and User3. This configuration makes single sign-on to discounts.com possible.
- At discounts.com, the administrator has configured the SAML Affiliate Agent to act as a consumer for smcompany.com, as in Example 3. An arrow connecting the two sites is not shown in the illustration. The administrator at discounts.com has also added configuration information about ahealthco.com so that the SAML Affiliate Agent can consume assertions from ahealthco.com for User2 and User3.

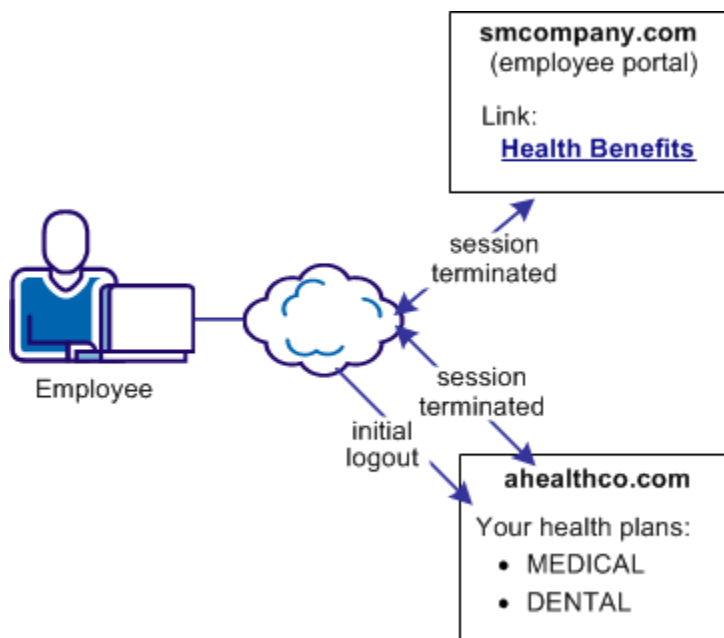
Use Case 5: Single Logout

In Use Case 5, an employee of smcompany.com authenticates at an employee portal, smcompany.com, and selects a link to view the health benefits at ahealthco.com. The employee is taken to the ahealthco.com website and presented with the health benefit information without logging in to the site.

After the employee logs out from ahealthco.com, the site wants to verify the termination of the user session at ahealthco.com and at smcompany.com. Terminating both sessions prevents an unauthorized employee from using the existing session to access resources at smcompany.com or to view benefits of the authorized employee.

Note: The initial logout can occur at smcompany.com and result in both sessions being terminated.

The following illustration shows the use case.



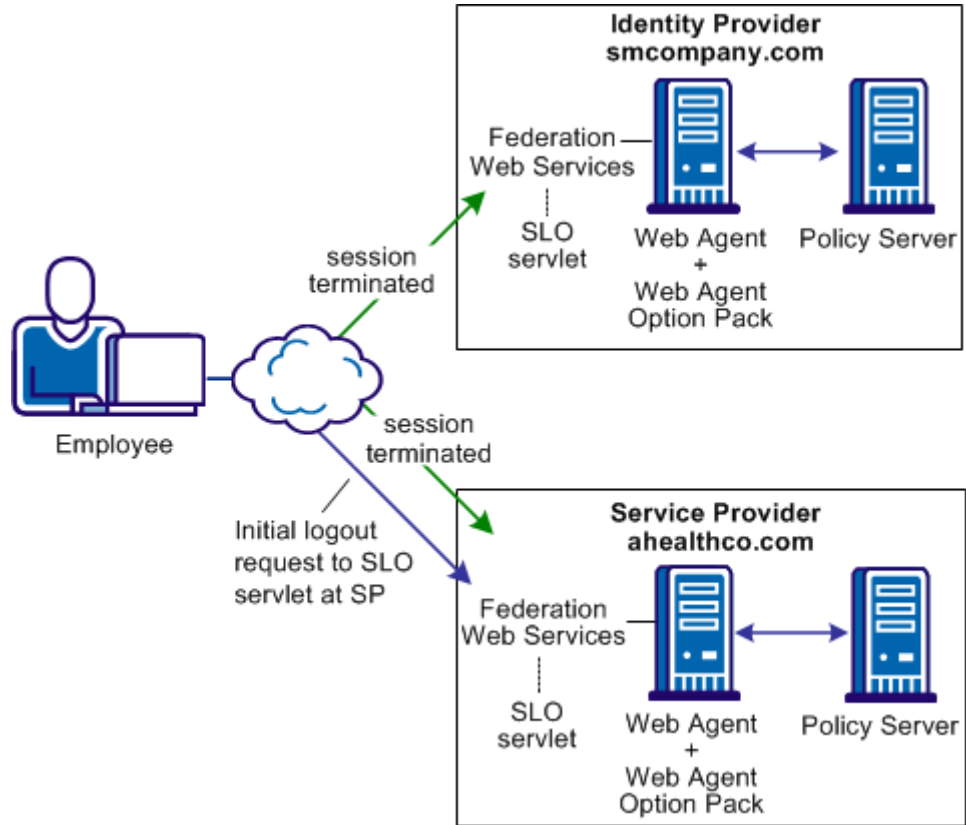
Solution 5: Single Logout (SAML 2.0)

Solution 5 illustrates how federation can be employed to solve [Use Case 5: Single Logout](#) (see page 48).

In this solution:

- smcompany.com is the Identity Provider
- ahealthco.com is the Service Provider that initiates the logout request.
- Single logout is enabled at the Identity Provider and the Service Provider.

The following figure shows the SiteMinder solution for single logout.



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

The sequence of events is as follows:

1. Employee performs single sign-on between smcompany.com and ahealthco.com. Smcompany.com places information about ahealthco.com in its session store. Ahealthco.com places information about smcompany.com in its session store.
2. After the employee has finished reviewing health benefits, the employee clicks a log out link at the Service Provider. The browser accesses the single logout servlet at the Service Provider.
3. The user session is terminated from the session store at the Service Provider.

Note: The termination does not remove the session from the session store; it sets the state to LogoutInProgress.

4. Based on information in the session store, the session is identified as one that an assertion from the Identity Provider, smcompany.com creates.

5. The browser is forwarded to the single logout servlet at smcompany.com, the Identity Provider, with the logout request message as a query parameter.
6. The Identity Provider invalidates the user session from all Service Providers that are associated with that user session, other than ahealthco.com, who initiated the logout request. After all Service Providers confirm the logout, the Identity Provider removes the user session from its session store.

Note: Other Service Providers are not identified in the illustration.

7. The Identity Provider returns a logout response message to ahealthco.com, the initiating Service Provider, and the user session is removed from the session store.
8. The user is finally sent to a logout confirmation page at ahealthco.com.

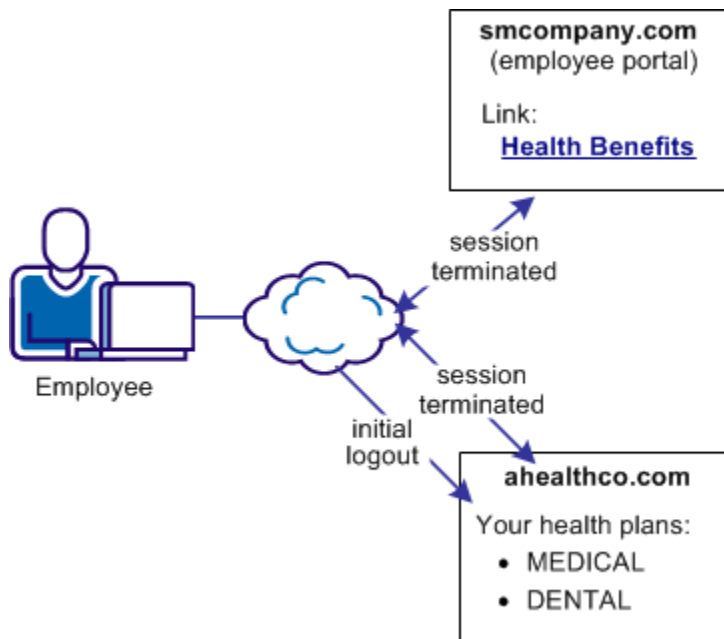
Terminating both sessions prevents an unauthorized employee from using the existing session to view benefits of the authorized employee.

Use Case 6: WS-Federation Signout

In Use Case 6, an employee of smcompany.com authenticates at the employee portal. The employee then selects a link to view health benefits at ahealthco.com. The employee is taken to the ahealthco.com website and presented with the health benefit information without having to sign on to the site.

When the employee logs out from ahealthco.com, ahealthco.com wants the user session at ahealthco.com and the session at smcompany.com terminated. Terminating both sessions helps ensure that an unauthorized employee cannot use the existing sessions to access resources at smcompany.com or to view benefits of the authorized employee.

The following illustration shows the use case.



Solution 6: WS-Federation Signout

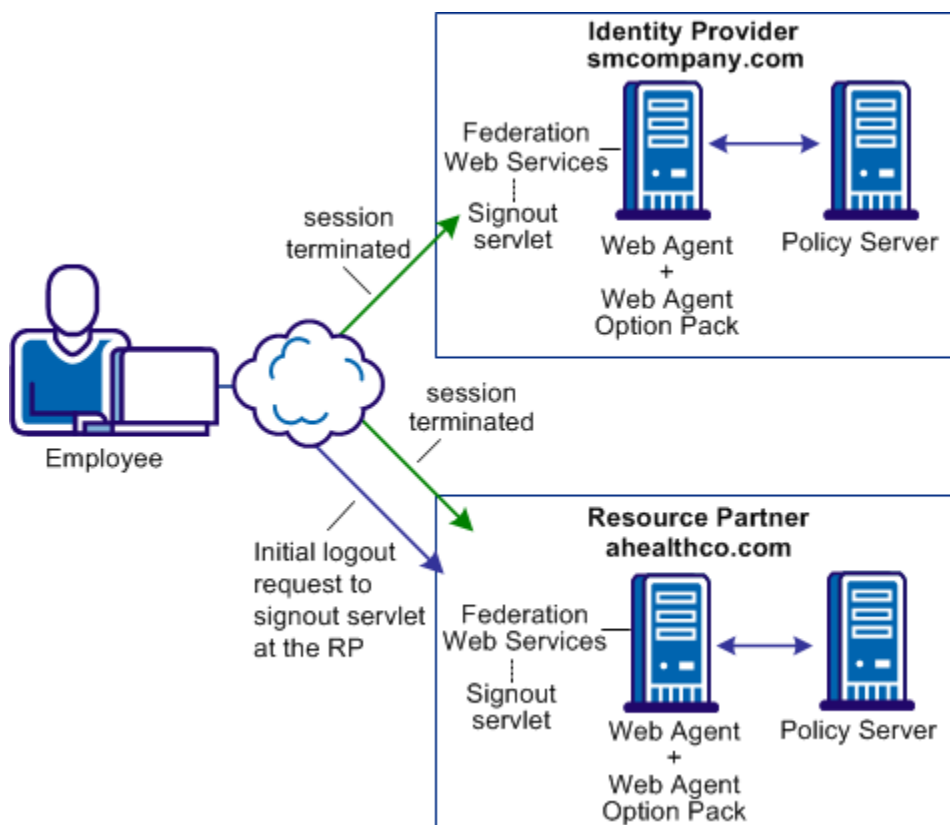
Solution 6 illustrates how federation solves [Use Case 6: WS-Federation Signout](#) (see page 51).

In this solution:

- smcompany.com is the Account Partner
- ahealthco.com is the Resource Partner that initiates the signout request.

WS-Federation signout is enabled at the Account Partner and the Resource Partner.

The following figure illustrates WS-Federation sign-out.



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

The sequence of events is as follows:

1. An employee authenticates at smcompany.com and then access their health benefits at ahealthco.com without having to log in. Federated single sign-on is configured between smcompany.com and ahealthco.com. During the transaction, smcompany.com places information about ahealthco.com in its session store. Ahealthco.com places information about smcompany.com in its session store.
2. The employee the health benefit information and clicks a logout link at ahealthco.com. The link calls the signout servlet at smcompany.com.
3. The session of the user is terminated from the session store of the Account Partner. All references to Resource Partners for that user are also removed from the session store.

4. The Account Provider retrieves a SignoutConfirm JSP page, which includes a Signout Cleanup URLs for each Resource Partner.

The SignoutConfirm page generates a frame-based HTML page with each frame containing a signoutcleanup URL for each Resource Partner that is associated with the user session.

5. The browser of the user then accesses the signout Cleanup URL at ahealthco.com and the session of the user is removed from the session store.
6. The browser of the user is finally sent back to the Account Partner.

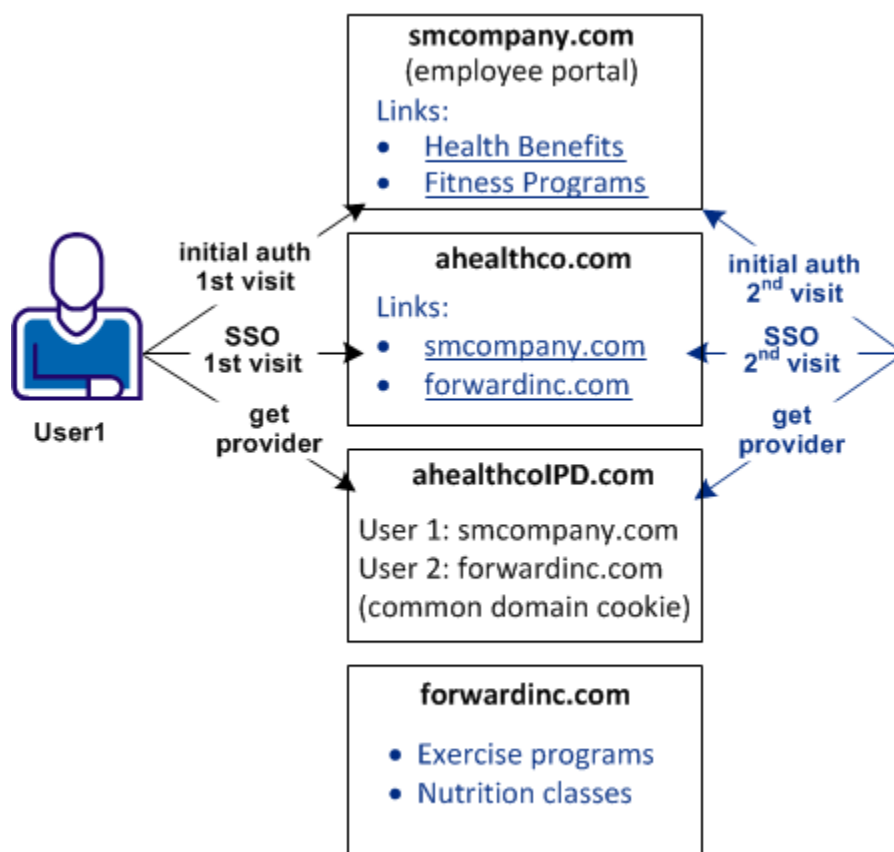
Steps 4-6 are repeated for each Resource Partner simultaneously for complete signout for that user session.

Use Case 7: Identity Provider Discovery Profile

In Use Case 7, several companies contract health benefits from ahealthco.com. When a user requests logs on to ahealthco.com to view their health benefits, ahealthco.com must determine which Identity Provider it sends an authentication request to for a particular user.

IdP discovery is useful in federated networks that have more than one partner providing assertions. It provides a dynamic way for a Service Provider to determine which Identity Provider it sends authentication requests for a particular user.

The following illustration shows a network with the Identity Provider Discovery profile in use.



A user arrives at ahealthco.com. This health provider determines where to send the authentication request. For User1, smcompany.com is where this user authenticates, so this company is set in the common domain cookie at ahealthco.com. For another user, forwardinc.com is an Identity Provider where a user authenticates. Forwardinc.com is set in the common domain cookie at ahealthco.com also.

A prior business agreement between the sites in this network exists so that all sites interact with the Identity Provider Discovery service.

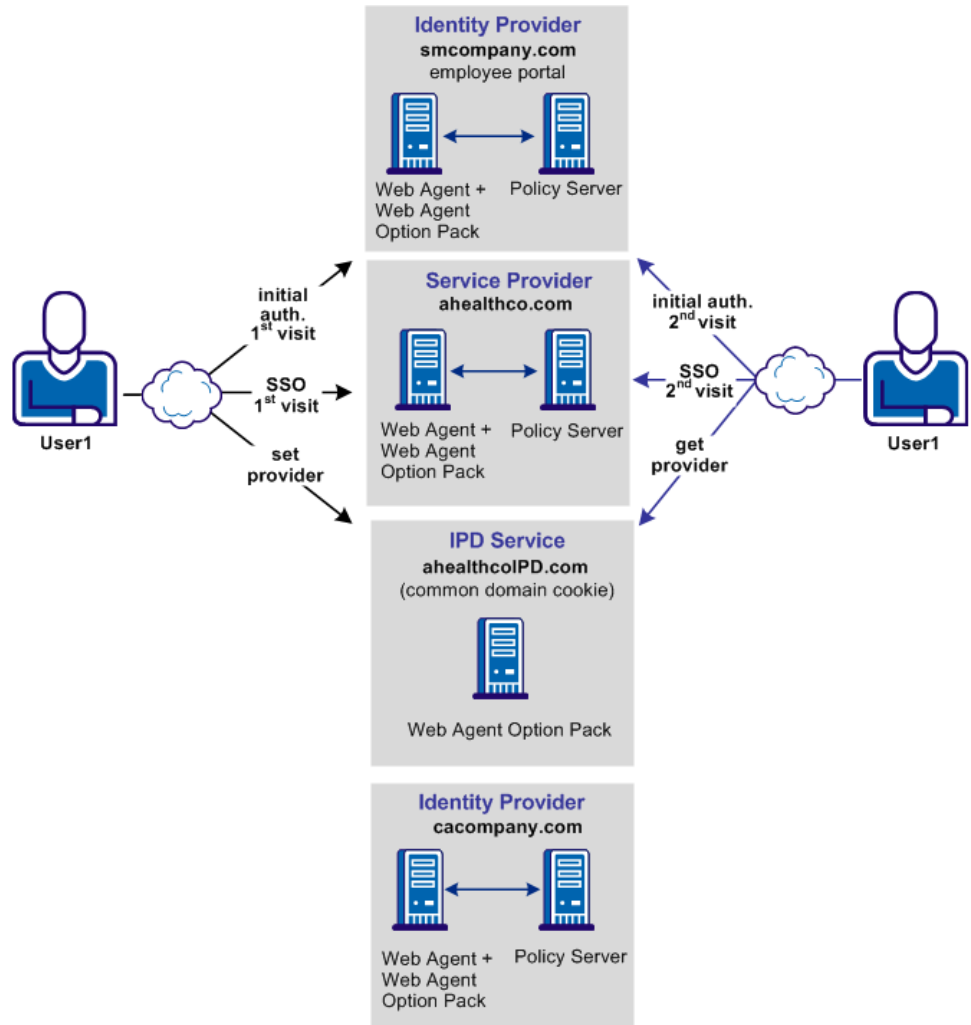
Solution 7: Identity Provider Discovery Profile (SAML 2.0)

Solution 7 illustrates how SiteMinder Federation Security Services can solve [Use Case 7: Identity Provider Discovery Profile](#) (see page 54).

In this solution:

- smcompany.com issues assertions for User 1 and has ahealthco.com configured as its Service Provider.
- ahealthco.com is the Service Provider for smcompany.com and cacompany.com. This site has a SAML 2.0 authentication scheme that is configured for each of these Identity Providers. This site enables single sign-on.
- ahealthcoIPD.com is the Identity Provider Discovery Service for ahealthco.com. The Federation Web Services application, which is installed with the Web Agent Option Pack, provides the IPD service which can read the common domain cookie. This common domain cookie includes all relevant Identity Providers for ahealthco.com.
- cacompany.com is another Identity Provider where users other than User1 can log in.

The following illustration shows the federated network for this solution.



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

The sequence of events is as follows:

1. User 1 initially authenticates at smcompany.com and then logs in to ahealthco.com without having to reauthenticate.

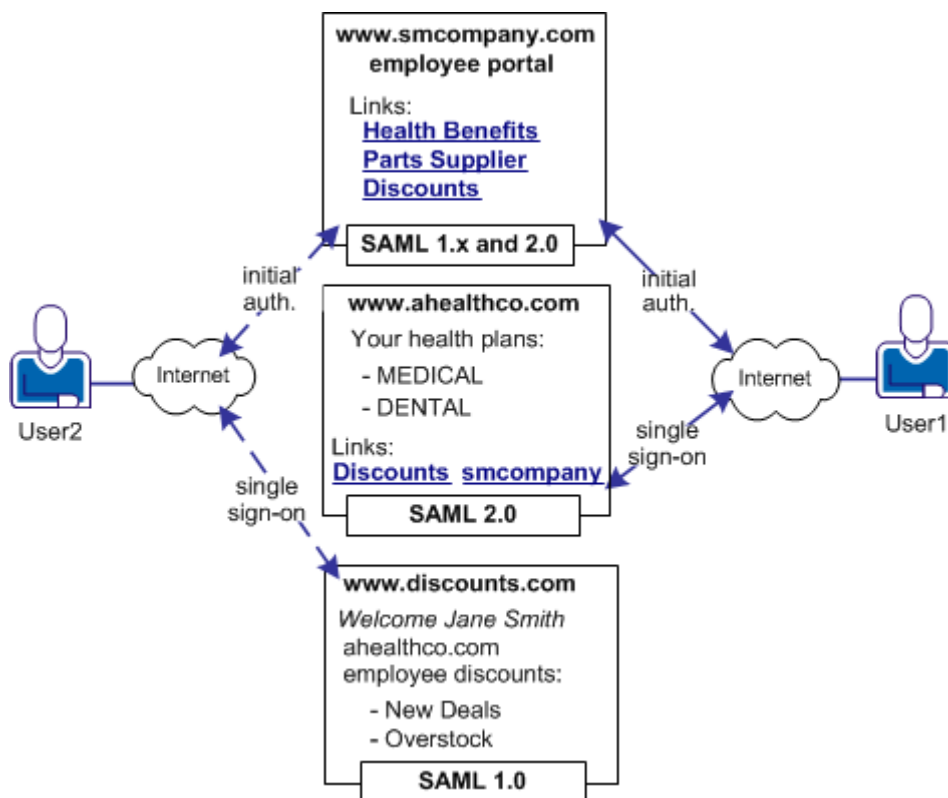
An agreement exists between smcompany.com and ahealthcoIPD.com to use ahealthcoIPD.com as the IPD service. During the initial authentication process, the Identity Provider URL of smcompany.com is written to the common domain cookie at the IPD service.

2. User 1, now successfully logged on to ahealthco.com, can look at the health benefits.
3. User 1 then comes to a site selection page at ahealthco.com. A common domain cookie is set for smcompany.com and ahealthco.com is configured to use the IPD service. As a result, ahealthco.com knows that the user previously logged in to smcompany.com. Therefore, ahealthco.com makes the appropriate links available to the user so that user can go back to smcompany.com to log in.

Use Case 8: Multi-protocol Support

In Use Case 8, smcompany.com issues assertions for ahealthco.com and discounts.com. Ahealthco.com uses SAML 2.0 for User1 to communicate between smcompany.com and ahealthco.com. Discounts.com uses SAML 1.0 for User2 to communicate between smcompany.com and discounts.com. The assertions must be suitable for the protocol that the SP uses to consume the assertion.

The following illustration shows the multiprotocol use case.



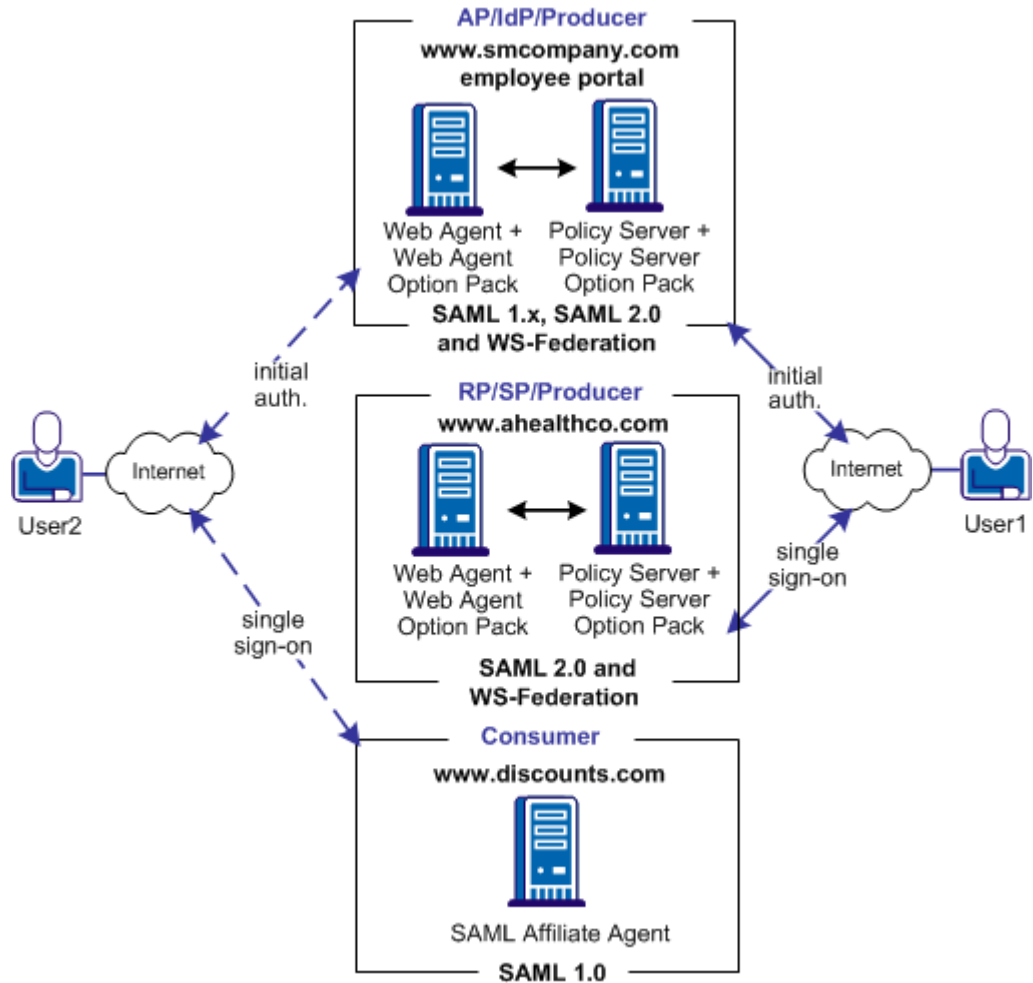
Solution 8: Multi-protocol Network

Solution 8 illustrates how SiteMinder Federation Security Services can solve [Use Case 8: Multi-protocol Support](#) (see page 58).

In this solution:

- For User 1,
 - smcompany.com is the SAML 2.0 Identity Provider for ahealthco.com. The partner, ahealthco.com, is included in an affiliate domain as a SAML 2.0 Service Provider.
 - ahealthco.com is where the SAML 2.0 authentication scheme is configured, and where smcompany.com is identified as the Identity Provider.
- For User 2,
 - smcompany.com is the SAML 1.0 producer for discounts.com, which is a SAML 1.0 consumer. This site uses the SAML Affiliate Agent, which can only consume SAML 1.0 assertions. This site cannot perform any authentication tasks.

The following illustration shows a SiteMinder federated network that implements multiprotocol support.



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

In this multiprotocol solution, smcompany.com can issue a SAML 2.0 assertion for User 1 to access resources at ahealthco.com. Additionally, smcompany.com can issue a SAML 1.0 assertion for User 2 to authenticate at discounts.com. Smcompany.com issues an assertion that is based on the session cookie that is set during initial authentication and determines the appropriate protocol for the assertion.

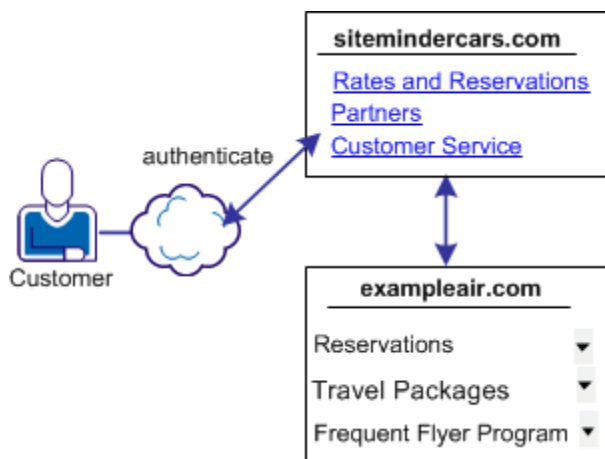
Configure the SAML Affiliate Agent at discounts.com. The smcompany.com is added to its producer information settings in its AffiliateConfig.xml configuration file so that it accepts SAML 1.0 assertions from this site.

Use Case 9: SAML 2.0 User Authorization Based on a User Attribute

In Use Case 9, sitemindercars.com is a car rental service.

A customer of sitemindercars.com logs in and authenticates at sitemindercars.com, then clicks a link to get a quote for a car rental. The customer has a customer profile at this site that includes the frequent flyer number of the customer with exampleair.com. The frequent flyer number of the customer miles determine a certain status level at sitemindercars.com, which offers the customer discounts on car rentals.

The following illustration shows this use case.

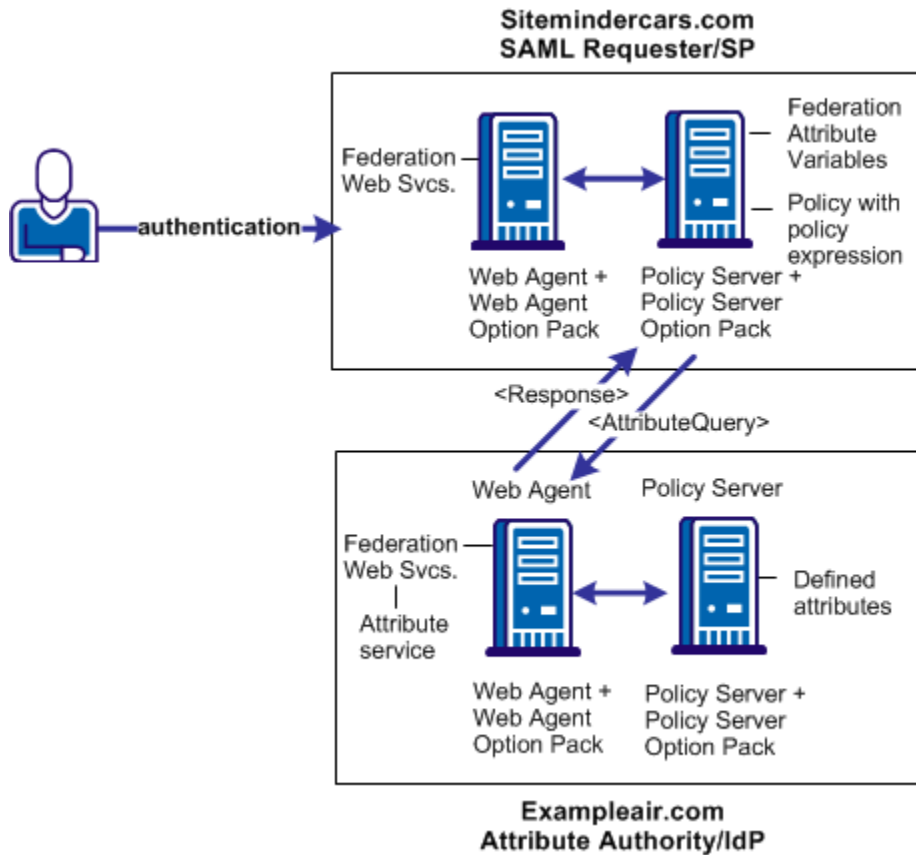


SiteMindercars.com wants to authorize its customers and present the appropriate discount information. The discount information is based on the frequent flyer number of the customer. The company does not want customers having to log in and authenticate at exampleair.com.

Solution 9: SAML 2.0 User Authorization Based on a User Attribute

Solution 9 shows how SiteMinder Federation Security Services can be deployed at sitemindercars.com and exampleair.com to solve [Use Case 9: SAML 2.0 User Authorization Based on a User Attribute](#) (see page 61).

Note: This solution is only for SAML 2.0.



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

SiteMinder is deployed at both sites. The Web Agent with the Web Agent Option Pack is installed on one system and the Policy Server with Federation Security Services on another system. The installations are the same for both sites.

Sitemindercars.com is a Service Provider acting as a SAML Requester. When a customer logs in at this site, the sequence of events is as follows:

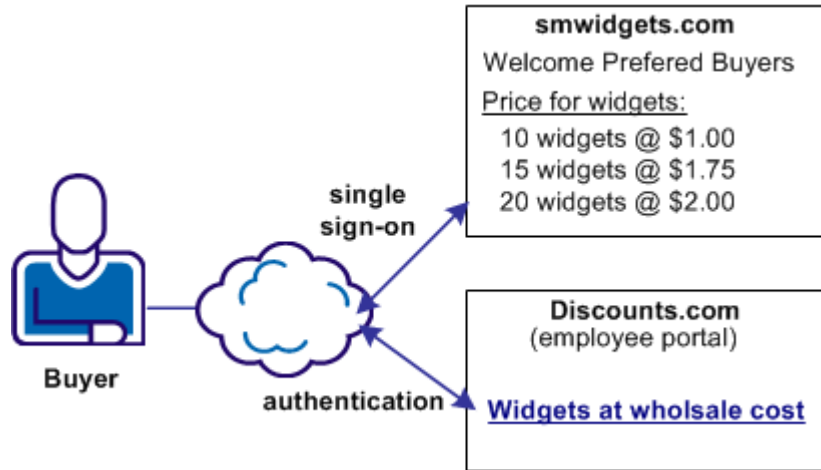
1. The user logs in to sitemindercars.com and the Web Agent provides the authentication.
2. When the user clicks a link to rent a car, the Web Agent first makes a request to the Policy Server at the local site.
3. The Policy Server evaluates the policy expression and identifies unresolved federation attribute variables. The Policy Server tries to resolve the variable by looking up the user in the user directory. The Policy Server looks in the directory associated with the policy protecting the requested URL.
4. The local Policy Server cannot resolve the user attribute variable. The Policy Server locates the NameID configuration for the Attribute Authority, Exampleair.com.
5. The Policy Server sends an <AttributeQuery> containing the NameID and the frequent flyer attribute to the Exampleair.com, which is acting as the Attribute Authority.
6. Exampleair.com returns a response to sitemindercars.com that contains an attribute assertion which includes the requested attribute.
7. The SAML Requester resolves the variables and evaluates the policy expression, returning authorization status to the Web Agent.
8. The Web Agent allows access to the appropriate resource.

Use Case 10: SAML 2.0 Single Sign-on with No Name ID at the IdP

In Use Case 10, discounts.com purchases widgets from smwidgets.com.

A buyer for discounts.com clicks on a link to access the latest widget price list at smwidgets.com. The buyer is taken to the smwidgets.com website and presented with the price list without having to log in to the discounts.com website.

The following illustration shows this use case.



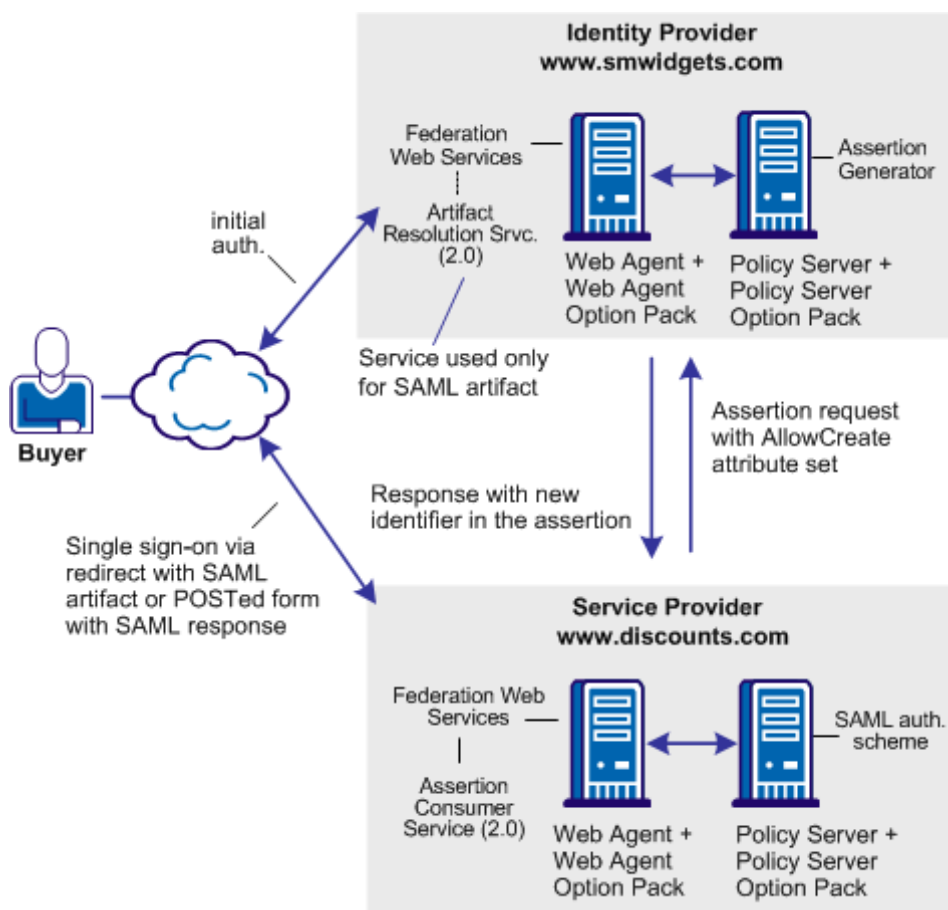
A buyer at discounts.com wants the price list at smwidgets.com. There are no buyer identities stored locally, so discounts.com wants to obtain an identity for the buyer at smwidgets.com. Discounts.com sends an authentication request to smwidgets.com. When smwidgets.com receives the request, it generates a unique persistent identity for the buyer and adds this identity to the assertion. Discounts.com uses this unique identifier to authenticate the user and allow the buyer access to the requested resource.

Solution 10: Single Sign-on with No Name ID at the IdP

Solution 10 shows how SiteMinder Federation Security Services can be deployed at smcompany.com and discounts.com to solve [Use Case 10: SAML 2.0 Single Sign-on with No Name ID at the IdP](#) (see page 63).

SiteMinder is deployed at discounts.com and smwidgets.com. A Web Agent and Web Agent Option pack is installed on one system, and the Policy Server with Federation Security Services is installed on another system.

In the following illustration, smwidgets.com is acting as the Identity Provider and discounts.com is acting as the Service Provider.



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

For single sign-on between the two sites, where there is no federated user identity at the Identity Provider, the sequence of events is as follows:

1. The user clicks on a link at discounts.com to proceed to the target site. This link initiates a call to the local Policy Server to generate an authentication request. In this request, an optional attribute named AllowCreate has been included, based on the configuration of the SAML 2.0 authentication scheme at the Service Provider.
2. The Federation Web Services application at the local Web Agent redirects the request to the Single Sign-on service at the IdP, smwidgets.com.

3. The request is then forwarded to the IdP Policy Server, which generates an assertion. During assertion generation, the Policy Server searches for the attribute associated with the user requesting access. For example, the telephone number attribute can be requested as the value of the Name ID.

If the Policy Server cannot find a value for the telephone number attribute, it verifies its configuration for the AllowCreate option. If this option is configured, the Policy Server searches the authentication request from the Service Provider to see if the AllowCreate option exists.

If the Allow/Create feature is enabled at both sites, the Policy Server generates a new identifier for the user attribute. The Policy Server places that identifier in its user store.

Note: The identifier is the value of the user attribute.

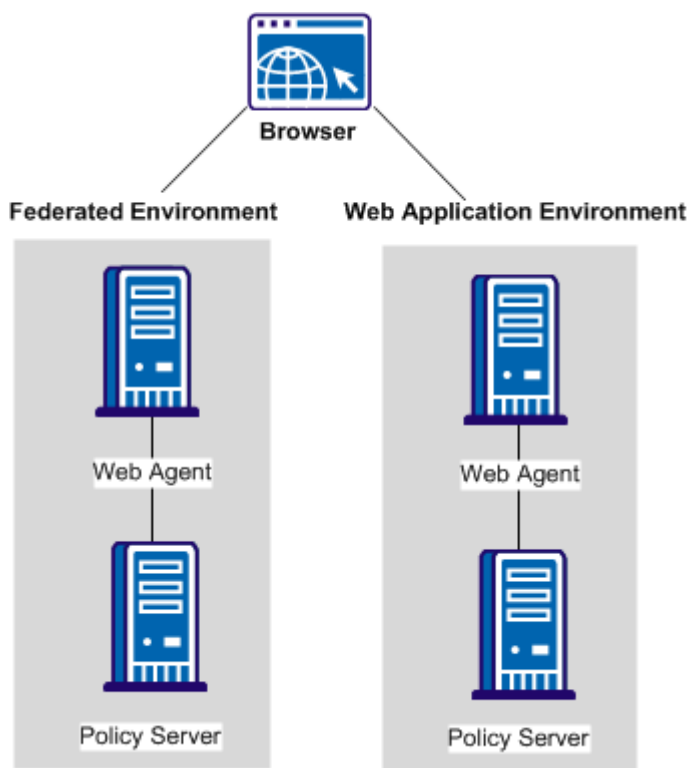
4. The assertion is returned in a response message to the IdP Web Agent (FWS). The IdP returns a form to the browser containing the response, the Assertion Consumer URL, and the javascript to submit the form.
5. The form is posted to the Assertion Consumer Service at the Service Provider. The Service Provider uses the response message to log in to the Policy Server, using the response as credentials.
6. The Service Provider at smwidgets.com validates the credential by looking for the attribute in its user store. Assuming that it finds the user, the user is logged in by the SAML authentication scheme.
7. The SP Web Agent creates an SMSESSION cookie for the smwidgets.com domain. The Agent places the cookie in the browser and redirects the user to the target destination.

Use Case 11: SAML Artifact SSO Using Security Zones

In use case 11, CompanyA, the producer site, wants to protect Web Agent applications and federated partner resources. The protocols that CompanyA uses for federated single sign-on are the SAML 2.0 artifact profile and SAML 2.0 single logoff.

For federated resources, a persistent user session is required because the SAML artifact profile stores assertions in the session store at the producer-side Policy Server. Consequently, calls are made to the session store to retrieve the assertion, impacting performance.

The following illustration shows a producer site that combines a federated environment and a web application environment.

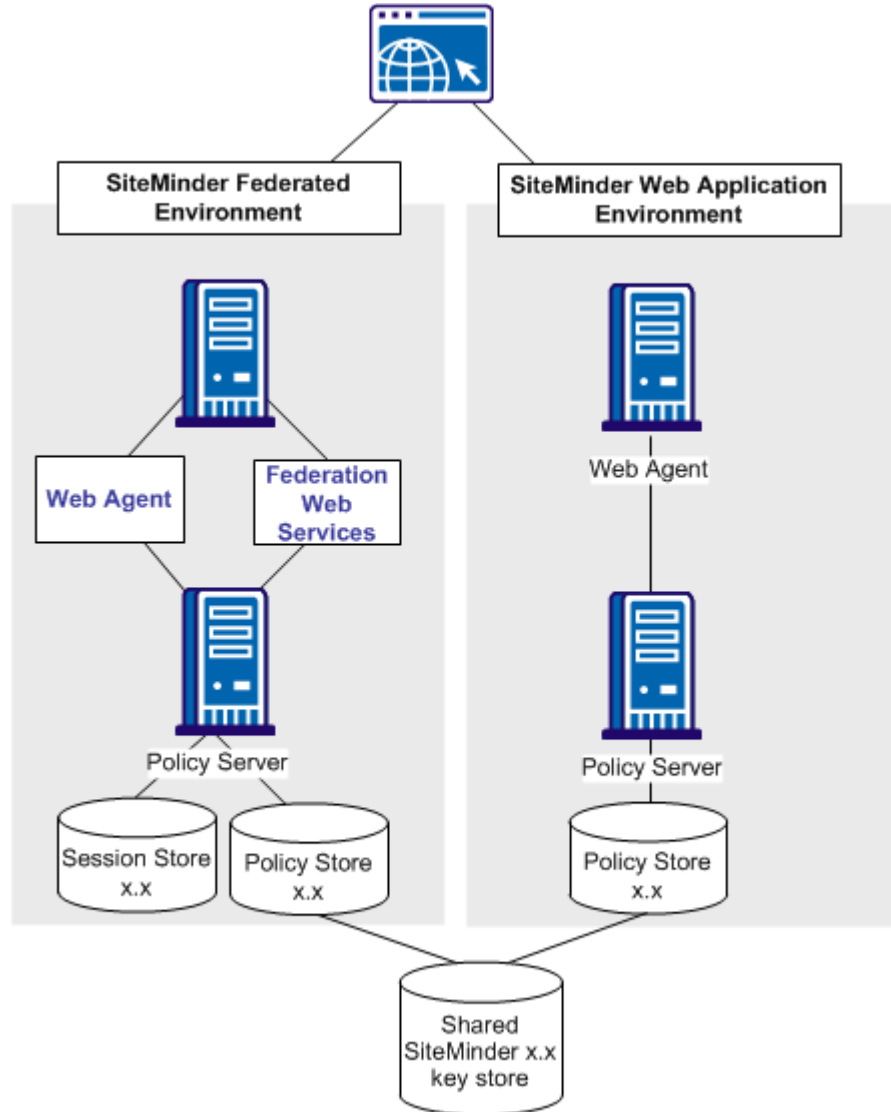


Solution 11: SAML Artifact SSO Using Security Zones

Solution 11 illustrates how you can set up a parallel web application and federation environments to solve Use Case 11.

A security zone is a segment of a single cookie domain, which is used as a method of partitioning applications. You can assign different security requirements to each zone. Producer-side Web Agents that protect requested federated resources enforce security zones. SiteMinder security zones eliminate the need for a persistent user session to be associated with every request for HTTP-Artifact protected applications.

The following figure illustrates a deployment that uses two different SiteMinder environments at a single asserting party. One SiteMinder environment is for federation functionality and the other is for web application protection.



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

The figure reflects the following set up:

Web application environment

Agent Configuration Object or Local Configuration File	Trusted Security Zones	Cookies Read by the Web Agent for the Zone
DefaultAgent	SM (default)--primary zone	The DefaultAgent configuration enables the Web Agent to read and write the default session cookie, SMSESSION.

Federation Environment

Agent Configuration Object or Local Configuration File	Trusted Security Zones	Cookies Read by the Web Agent for the Zone
FedWA used by the Web Agent	FED--primary zone SM--additionally accepted zone	The FedWA configuration enables the Web Agent to read and write SESSIONSIGNOUT cookies, and read SMSESSION cookies.
FedFWS used by the FWS application	FED--primary zone only	Configures the FWS to read and write SESSIONSIGNOUT cookies.

All resources protected in the web application environment use non-persistent user sessions. As a result, when users are authenticated and authorized, the SMSESSION cookie contains a non-persistent user session specification. The non-persistent session specification helps ensure that requests to web applications do not incur the performance penalty of calling the session store.

When the Web Agent in the federation environment receives a request, this request is directed to the Authentication URL to establish a user session. The user making the request already has an SMSESSION cookie from the prior authentication in the web application environment. However, the user has no SESSIONSIGNOUT cookie.

The Web Agent in the federation environment writes a SESSIONSIGNOUT cookie. The SESSIONSIGNOUT cookie has a persistent user session specification and it uses the same session ID as the SMSESSION cookie. This persistent user session protects the Authentication URL, which authenticates users federating to a partner site.

The Web Agent in the federation environment reads the SMSESSION cookie and writes a SESSIONSIGNOUT cookie in accordance with the security zones associated with the FedWA configuration.

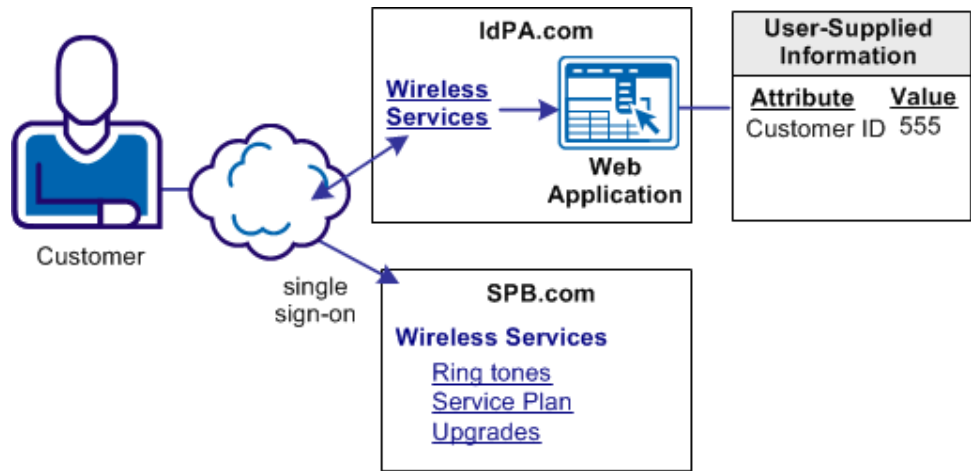
The Federation Web Services application in the federation environment reads the SESSIONSIGNOUT cookie. Because the cookie contains a persistent user session, a call to the session store is not necessary. The Federation Web Services application successfully processes the federation request that requires a persistent user session.

Use Case 12: SAML 2.0 SSO Using Attributes from a Web Application

In use case 12, an Identity Provider, IdPA.com, wants to include web application attributes in an assertion. This use case is only applicable to SAML 2.0 deployments.

For this use case, single sign-on can be initiated at the Identity Provider or the Service Provider. The profiles that IdPA.com uses is SAML 2.0 (POST and Artifact) and WS-Federation.

The following figure shows an example of this use case.



IdP-initiated Single Sign-on with Web Application Attributes

IdPA.com has a web application that allows access to protected resources at its business partner SPB.com. When the customer logs in at IdPA.com, they select a link for the business partner. The user is sent to the web application, where they are prompted to enter a customer ID. IdPA.com must send this information to SPB.com so that the customer is permitted access to the requested resource.

SP-initiated Single Sign-on with Web Application Attributes

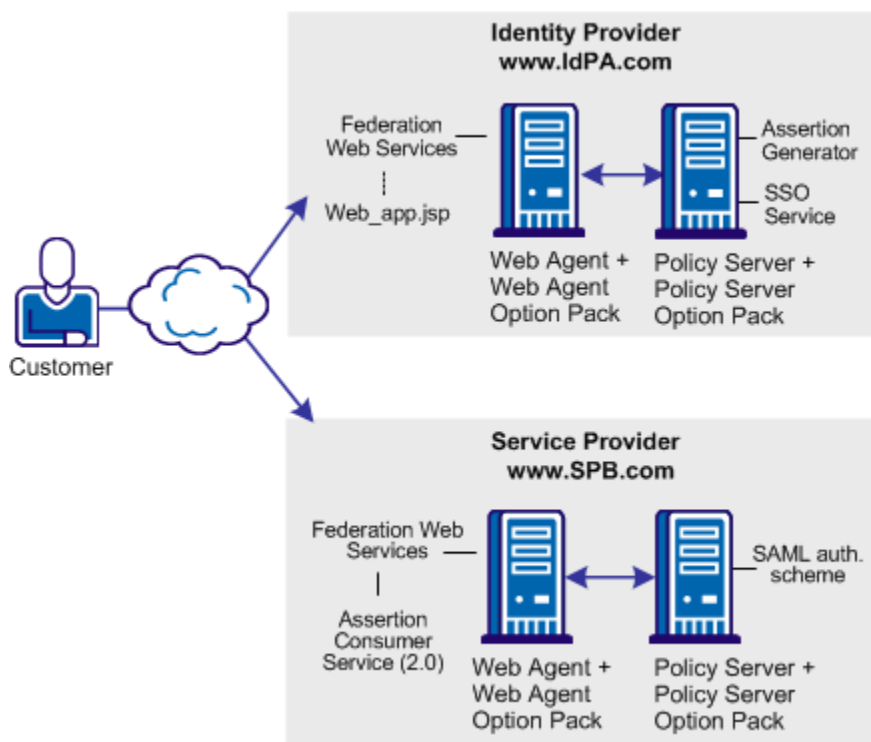
A customer selects a link at SPB.com, the Service Provider. This link is for protected resources, so the customer is redirected to IdPA.com to be authenticated. After the customer successfully authenticates at IdPA.com, the IdP redirects the customer to the web application where the customer provides specific user information. Upon submitting the information, the customer is returned to SPB.com to complete single sign-on for the requested resource.

Solution 12: SSO with Attributes from a Web Application

Solution 12 shows how SiteMinder Federation Security Services can be deployed at IdPA.com and SPB.com to solve [Use Case 12](#) (see page 70).

SiteMinder is deployed at both sites. At each site, the Web Agent and the Web Agent Option pack are installed on one system, and the Policy Server is installed on another system.

In the following illustration, IdPA.com is the Identity Provider and SPB.com is the Service Provider and single sign-on is initiated at the Identity Provider.



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

For IdP-initiated single sign-on, the sequence of events is

1. At the IdP, the user clicks the web page link and *one* of the following actions occurs:

- The user is directed to Single Sign-on (SSO) service from IdPA.com. This service recognizes that the user does not have a session. The user is redirected to the IdP and is prompted to log in. After successful login, the user is redirected back to the SSO service. An application URL defined for the SSO service instructs the SSO service to send the user to a custom web application.

Important! When the user goes to the SSO Service first, several query parameters (SPID, ProtocolBinding, RelayState) are included with the original SSO request. The SSO service groups this query data into one query parameter named SMPORTALSTATE, and then redirects the user (through a GET) to the web application.

- The user logs in and is taken directly to the web application.

Note: After the user is locally authenticated at the IdP, the user is never redirected to the Authentication URL if the session is valid.

2. The web application prompts the user supplies the requested information. These attributes are posted to the SSO Service.

Important! The web application must maintain and POST the SMPORTALSTATE query parameter and the collected attributes back to the SSO Service.

3. The SSO service processes the SAML request and unpacks the data from the SMPORTALSTATE parameter. The service takes this data with the attributes from the web application and passes all the POST data to the assertion generator.

4. The Assertion Generator creates the assertion.

Important! The SSO Service makes all the attributes *available* to the Assertion Generator. Write and configure the assertion generator plug-in to add the attributes to the assertion.

5. After the IdP generates the assertion, the IdP redirects the user to the Assertion Consumer Service at the Service Provider. The Service Provider processes the assertion.

6. The user gains access to the requested resource at the Service Provider.

For SP-initiated single sign-on:

1. At the SP, the user clicks on a link and an AuthnRequest is sent to the Single Sign-on (SSO) service at the Identity Provider.

Note: In the SP-initiated single sign-on, the request must arrive at the SSO service directly from the SP, as the SAML specification requires. The user cannot go directly to the web application.

2. At the IdP, the SSO service recognizes that the user does not have a session. The user is redirected to the Authentication URL to authenticate and establish a session. After the user establishes a session, the IdP redirects the user back to the SSO service. An application URL defined for the SSO service instructs the SSO service to return the user to a custom web application.

Important! When the user is directed to the SSO Service, several query parameters (SPID, ProtocolBinding, RelayState) are included with the original request. The SSO service groups this query data into one query parameter named SMPORTALSTATE, and then redirects the user (through a GET) to the web application.

3. The web application prompts the user to supply the requested information. These attributes are posted to the SSO Service.

Important! The web application must maintain and POST the SMPORTALSTATE query parameter and the collected attributes back to the SSO Service.

4. The SSO service processes the SAML request and unpacks the data from the SMPORTALSTATE parameter. The service takes the data and the attributes from the web application and passes all the POST data to the assertion generator.
5. The IdP generates the assertion and includes all the attributes. The IdP redirects the user to the Assertion Consumer Service at the Service Provider, where the assertion is processed.

Note: Write and configure an assertion generator plug-in to add the attributes to the assertion.

6. The user gains access to the requested resource at the Service Provider.

Configure SSO with Attributes from a Web Application

Configuring single sign-on based on attributes from a web application, requires specific steps.

Follow these steps:

1. Create a custom web application for the IdP in your network. This custom application can prompt the user for as many attributes as required. Conversely, the application can supply standard attributes and not prompt the user for any information. How attributes are gathered is entirely dependent on how the custom application is written.

Important! For IdP-initiated single sign-on, if the user is directed to the web application before the SSO service, the web application must include the parameter **AllowApplicationPost=yes**. The SSO service accepts the post as long as the application includes the AllowApplicationPost parameter.

The SiteMinder Web Agent Option Pack comes with sample JSP applications that you can use as a basis for your custom web application. The path to the sample JSP applications is: *web_agent_home/affwebservices/*. The sample applications are:

sample_application.jsp

This sample application can be used for IdP- or SP-initiated single sign-on. The user is first directed to the SSO Service and then sent to the custom web application. This application can be entered for the Application URL in the Service Provider Properties (SAML 2.0) dialog or the Resource Provider Properties (WS-Federation) dialog.

unsolicited_application.jsp

This sample application can be used for IdP-initiated single sign-on. The user is sent directly to the web application and not to the SSO Service. The application assumes that the user is already authenticated at the Identity Provider.

Note: This file shows how to use the AllowApplicationPost parameter in an application.

2. (Optional) If the user is initially directed to the IdP SSO service:
 - a. Specify an Application URL in the SAML 2.0 authentication scheme.
 - b. Configure the Assertion Generator plug-in to add the attributes to the assertion.
3. (Optional) If the user is sent directly to the custom web application from the IdP, you do not have to provide a value for the Application URL parameter. However, write and configure the assertion generator plug-in to work with SiteMinder. See the *Programming Guide for Java* for information about creating an assertion generator plug-in.

Note: The order of the procedure steps is provided as a guideline. You can perform these steps in a different order.

More information:

[Enable the Assertion Generator Plug-in \(SAML 2.0\)](#) (see page 341)

Use Case 13: SSO with Dynamic Account Linking at the SP

In Use Case 13, the IdP, discounts.com, includes an attribute named buyerID that identifies a particular user and is included in an assertion. When the assertion is sent to the Service Provider, smwidgets.com, the same attribute does not exist in the user record at the Service Provider. The Service Provider must create an attribute in the appropriate user record so that the user can authenticate and gain access to the protected resource.

An employee of discounts.com selects a link to access the latest price list on widgets at smwidgets.com. The employee logs in with the name and buyer ID.

The following illustration shows this use case.



The identity that is based on the buyer ID of the user is created at discounts.com and placed in the assertion. The buyer ID value is entered as the NameID in the assertion. However, there is no mapped identity at smwidgets.com for the buyer ID. The administrator at the Service Provider establishes a mapping. The mapping has to use dynamic account linking so that smwidgets.com can authenticate the employee and can allow the employee access to the price list.

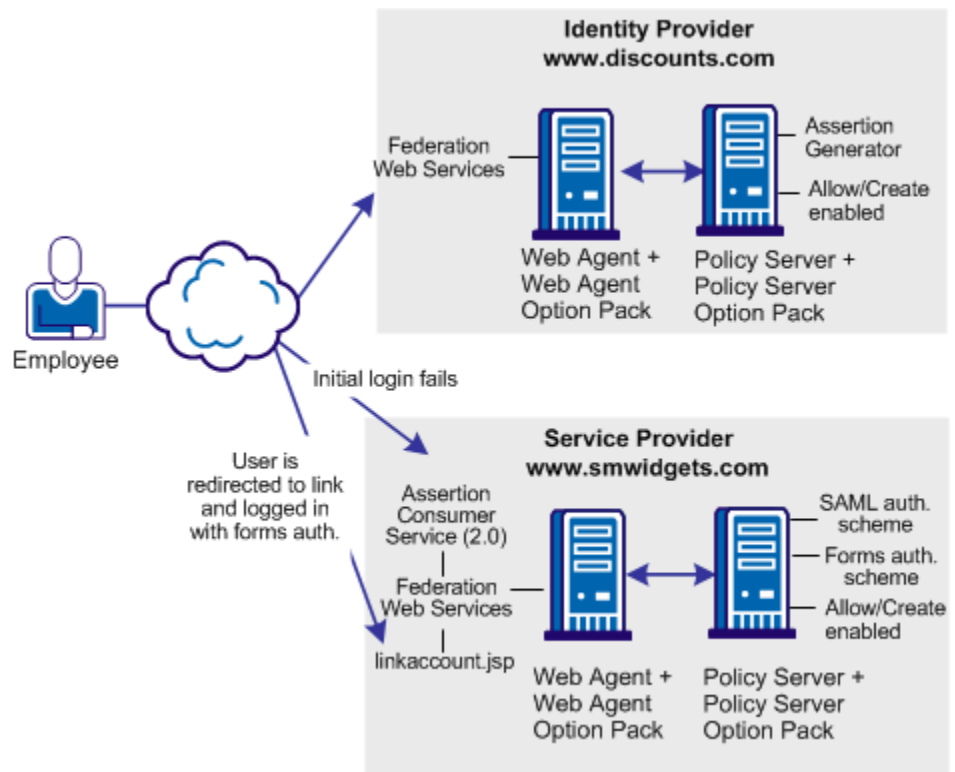
Solution 13: SAML 2.0 SSO with Dynamic Account Linking at the SP

Solution 13 shows how SiteMinder Federation Security Services can be deployed at IdPA.com and SPB.com to solve [Use Case 13](#) (see page 75).

Note: Dynamic account linking is only supported with SAML 2.0.

SiteMinder is deployed at both sites. Each site has a Web Agent and Web Agent Option Pack installed on one system, and the Policy Server on another system.

The following illustration shows single sign-on with dynamic account linking at the Service Provider.



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

The order of events for this solution is

1. The employee initially logs in and authenticates at discounts.com. Discounts.com creates an assertion for the employee. Discounts.com posts the assertion (POST binding) or redirects the user with an artifact (artifact binding) to the Assertion Consumer Service at smwidgets.com. This assertion includes an attribute named buyerID.
2. The Assertion Consumer Service at smwidgets.com tries to authenticate the user with the SAML authentication scheme. However, the buyerID attribute of the employee does not map to a local user record so the authentication fails.
3. As part of the SAML authentication scheme at the SP, a redirect URL is defined, which points to the directory *web_agent_home/affwebservices/linkaccount.jsp*. The employee is redirected to this URL.

Note: The linkaccount.jsp file must be part of a protected realm. The default location for this file is `http://sp_home/affwebservices/public/`. Copy the file from this location to a protected realm.

4. A Web Agent that authenticates the local user with the forms authentication scheme protects this linkaccount.jsp URL. After a successful authentication, a SiteMinder session at smwidgets.com is established and an SMSESSION cookie is placed in the browser of the employee.
5. The linkaccount.jsp gets loaded in the browser and the user sees a message to permit the link to the SP account. Click on the button to permit the account linking.
6. The user is redirected to the Assertion Consumer Service, where the browser of the employee presents the SMSESSION cookie with the assertion.
7. The Assertion Consumer Service extracts the NameID from the assertion and inserts the NameID value into a newly created buyerID attribute. The buyerID attribute is in the existing user record of the employee. The Assertion Consumer Service knows which user record to map because the UserDN in the SMSESSION cookie identifies the user.

The search specification configured in the SAML 2.0 authentication scheme indicates which attribute is mapped to the NameID. In this case, the search specification is `buyerID=%s`.

8. After the attribute is mapped, the SAML authentication scheme authenticates the user that is based on the assertion and establishes a new user session.

The next time that the same user presents an assertion with the buyer ID, the user successfully gains access to the requested resource.

Configure SAML 2.0 SSO with Dynamic Account Linking at the SP

Configure several components at the Service Provider to enable SAML 2.0 single sign-on with dynamic account linking:

- AllowCreate feature
Enables the creation of attributes in an existing user store.
- Redirect URL
Sends the user to the linkaccount.jsp file when authentication fails. An authentication protects the redirect URL. The scheme requests the user to log in to create a SiteMinder session.
- Post Preservation at the Web Agent
Must be enabled at the Service Provider Web Agent.
- Search Specification
Indicates which attribute the NameID from the assertion replaces

Enable dynamic account linking for POST or Artifact single sign-on at the Service Provider

Follow these steps:

1. For the linkaccount.jsp file, do the following:
 - (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 URL.
Note: The accountlinking must be set to yes (accountlinking=yes).
The default location for this file is `http://sp_home/affwebservices/public/`.
 - Protect the linkaccount.jsp file with a SiteMinder forms authentication scheme, which supports POST-Preservation. The SAML response that contains the assertion is posted to the Assertion Consumer Service after the user has logged in locally at the Service Provider. Preserve the SAML response POST data during the entire local authentication process.
To protect resources with an authentication scheme, refer to information about authentication schemes in the *Policy Server Configuration Guide*.
2. Enable the Allow/Create feature at the Service Provider.
3. For the Web Agent at the Service Provider, set the POST Preservation parameter to yes. This setting enables the POST data from the SAML response to be preserved.

4. Configure a redirect URL that sends the user to the linkaccount.jsp file if authentication fails. Direct the user only to this file.

The redirect URL is part of the SAML 2.0 authentication scheme setup at the Service Provider.

Complete the following fields with the values shown:

Redirect URL for the User Not Found Status

`http://sp_home/protected_realm/linkaccount.jsp`

Example: `http://smwidgets.com/partner_resources/linkaccount.jsp`

The default location of the linkaccount.jsp file is

`http://sp_home/affwebservices/public/`. Copy the file from this directory to a directory that is configured as a protected realm.

Mode

HTTP POST

5. Configure a search specification for the SAML authentication scheme. For example, if the Name ID from the assertion replaces buyerID, the search specification would be `buyerID=%s`.

More information:

[Use Case 13: SSO with Dynamic Account Linking at the SP](#) (see page 75)

[Allow the Identity Provider to Assign a Value for the NameID](#) (see page 310)

[Permit the Creation of a Name Identifier for SSO](#) (see page 361)

[Look Up User Records for SAML 2.0 Authentication](#) (see page 354)

[Locate User Records for Authentication](#) (see page 429)

SiteMinder Administrative User Interfaces

Beginning with SiteMinder r12 SP1, there are two graphical user interfaces (UIs) that configure SiteMinder policy objects, as follows:

Administrative UI

The Administrative UI is a web-based administration console that is installed independent of the Policy Server. Use the Administrative UI to view, modify, and delete all Policy Server objects except those related to Federation Security Services. All federation-related configuration tasks should be handled using the FSS Administrative UI.

Federation Security Services Administrative UI (FSS Administrative UI)

The FSS Administrative UI is an applet-based application that is installed with the Policy Server. The federation-specific UI objects consist of affiliates (consumers, service providers, resource partners) and SAML authentication schemes that you configure to support federated communication between two partners.

The intent of the FSS Administrative UI is to let you manage SiteMinder Federation Security Services. If you are familiar with previous versions of the SiteMinder Policy Server User Interface, you will notice that all SiteMinder objects appear in the FSS Administrative UI. The only objects that do not appear are objects related to Enterprise Policy Management (EPM) and reports. You can use the FSS Administrative UI to manage the SiteMinder objects. If you need information while using the FSS Administrative UI, consult the FSS Administrative UI online help system.

Important! You must register each UI with the Policy Server. Registering the FSS Administrative UI with the Policy Server ensures that the communication between both components is FIPS-encrypted (AES encryption). For more information about registering a UI, see the *Policy Server Installation Guide*.

Chapter 2: Deploy Federation Using the Sample Application

This section contains the following topics:

[Legacy Federation Sample Application Overview](#) (see page 81)

[Legacy Sample Application Deployment](#) (see page 81)

[Sample Application Components](#) (see page 82)

[Prerequisites to Deploy the Sample Application](#) (see page 84)

[How To Run the Sample Application](#) (see page 85)

[Test Single Sign-on with the Sample Application](#) (see page 92)

[Test Single Logout with the Sample Application](#) (see page 93)

[Review Application-Generated SiteMinder Objects](#) (see page 94)

Legacy Federation Sample Application Overview

To become familiar with SiteMinder Federation Security Services, deploy the legacy federation sample application. The sample application automates all the federation setup tasks to accomplish SAML 2.0 single sign-on and single logout. After you run the sample application, look at the SiteMinder policy objects that the sample application creates. Also, examine the SiteMinder logs containing assertions. Finally, use the sample application objects as a basis for configuring your own federation environment.

Note: The Federation Security Services sample application only creates SAML 2.0 objects.

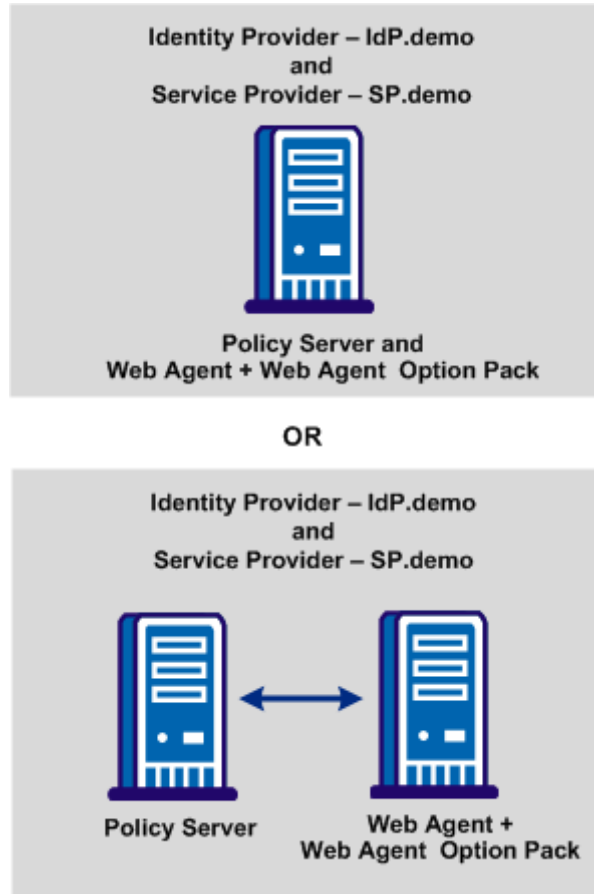
Legacy Sample Application Deployment

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

You can deploy the sample application in many ways. We recommend one of two ways:

- On one system acting as an Identity Provider (IdP) and a Service Provider (SP). In this instance, install all SiteMinder components (Policy Server, Web Agent, and Web Agent Option Pack) on one system.
- Two systems acting as an Identity Provider (IdP) and a Service Provider (SP). Install the Policy Server on one system and the Web Agent and Web Agent Option Pack on a second system.

The following illustration shows two deployments of the sample application.



Sample Application Components

The legacy sample application contains the following components:

FederationSample.conf

The FederationSample.conf file contains configuration settings that define the IdP and SP-side policy objects.

SetupFederationSample.pl Perl script

The SetupFederationSample.pl Perl script executes the federation sample application. This script creates the objects for the IdP and SP sites. The script also creates the necessary web pages to initiate single sign-on and single logout between the IdP and the SP. The script relies on the information in the FederationSample.conf file to operate.

Use the Perl interpreter included with the sample application to run the application.

Web pages to test single sign-on and single logout

The sample application installs two directories that contain template pages for testing SAML 2.0 single sign-on and single logout transactions. The directories, **idpsample** and the **spsample**, are installed in the directory *siteminder_home/siteminder/samples/federation/content*.

These directories are also copied to the default document root directory of the web server.

- IDP Pages

The IdP web pages are in the idpsample directory. These pages include:

index.jsp

Index.jsp is the first web page the user accesses at the IdP for IdP-initiated single sign-on. This page provides the link to the protected target resource at the sp.demo partner site. This page also provides a single logout link.

Note: The single logout link is displayed only if FSS is the IdP and an SMSESSION cookie is in the request headers.

SLOConfirm.jsp

SLOConfirm.jsp displays a message that the user has successfully logged out from idp.demo and sp.demo domains.

- SP Pages

The SP web pages are in the spsample directory. These pages include:

index.jsp

Index.jsp is the first web page the user accesses at the SP for SP-initiated single sign-on. This page provides a link to the protected target resource. This page also provides single logout link.

Note: The single logout link is displayed only if FSS is the IdP and an SMSESSION cookie is in the request headers.

target.jsp

Target.jsp, a protected page at the sp.demo partner site, is located in the */spsample/protected* directory. The SAML 2.0 authentication scheme protects this page. A user sees this page when single sign-on between the IdP and SP succeeds.

SLOConfirm.jsp

SLOConfirm.jsp displays a message that the user has successfully logged out from the idp.demo and sp.demo domains.

Prerequisites to Deploy the Sample Application

Before you run the sample application, satisfy the following requirements.

Deployment requirements (SiteMinder r12.0 SP3 components recommended):

- Install a Policy Server.
- Install a Web Agent.

The web server where you install the Web Agent does not require an SSL port.

- Install the Web Agent Option Pack. The Web Agent Option Pack requires a supported application server.

If you install the Policy Server, Web Agent, and Web Agent Option Pack on one system, we recommend using ServletExec as the application server. To install the Web Agent Option Pack and deploy Federation Web services on an application server, see the *Web Agent Option Pack Guide*.

- Install an LDAP or ODBC user directory.

On the Policy Server system:

- Point the Policy Server to the user directory that you set up.
- Configure a policy store.
- Configure and enable a session store to use the sample application for testing HTTP-Artifact single sign-on and single logout. Set up a separate session store repository from the policy store repository. The session store can be CA Directory (LDAP) or an ODBC database.

For instructions on configuring a session store, see the *Policy Server Administration Guide*.

On the web agent system:

- (Optional) Enable logging in the `LoggerConfig.properties` file. This file is in the directory `web_agent_home/affwebservices/WEB-INF/classes`. Enabling trace logging generates the `FWSTrace.log` file. The trace log lets you view the assertion after you test single sign-on.

Note: If your deployment uses only one system, install all components on that one system. If your deployment has more than one Policy Server and more than one Web Agent, complete the prerequisites on all relevant systems.

Verify that the Policy Server and Web Agent are configured properly and that you can protect a resource.

Important! Core SiteMinder must function properly to run the sample application successfully.

How To Run the Sample Application

After you complete the necessary prerequisites, set up your environment to run the sample application.

On the Policy Server system

1. Set your command line path.
2. From the FSS Administrative UI, configure a web agent and name it **agent**.
3. Modify the FederationSample.conf file for your environment.
4. Run the SetupFederationSample.pl Perl script to run the sample application.

On the Web Agent system

1. Copy the idpsample and spsample directories to the Web Agent system.
2. Add a virtual directory for the idpsample and spsample directories on the Web Agent system.
3. Edit the host file on the Web Agent system and any other system from where you plan to access the application from a browser.

After your setup is complete on all systems, test single sign-on and single logout.

Set the Path Variable on the Policy Server System

On the system where the Policy Server resides, set the path variable. You can set the Path as a command line path, which affects only the local command prompt.

Follow these steps:

1. Open a command window.
2. Enter the following command:

```
set path=%NETE_PS_ROOT%\cli\bin;%NETE_PS_ROOT%\cli\lib;%path%
```

Important! Verify that the Perl binary bundled with the Policy Server is the first or only such binary in the PATH. Invoke the bundled Perl binary, not another Perl script.

Note: You can also set the Path in the system environment variable.

Configure a Web Agent in the FSS Administrative UI

Create a Web Agent and name it agent.

Follow these steps:

1. Log in to the FSS Administrative UI.
2. From the System tab, select Edit, Create Agent.
3. Create a web agent object.
4. Enter **agent** in the Name field.
5. Click OK to save the object.

Modify the FederationSample.conf File

The FederationSample.conf file holds the settings for the local environment, such as your web server port and user directory. This file also contains one setting for the partner site.

The application uses the FederationSample.conf file to create Identity Provider and Service Provider policy objects.

To modify the FederationSample.conf file

1. Go to *policy_server_home/samples/federation*.
2. Open the FederationSample.conf file.
3. Modify the [file settings](#) (see page 86).
4. Save the file.

FederationSample.conf Settings

Configure all the settings in the FederationSample.conf file.

The settings are as follows:

USER_DIRECTORY

Specifies the name of an existing user directory object in the FSS Administrative UI. This directory must contain at least one user entry. If no value is specified for this setting, the sample application script reads the user directory information from the policy store, provided only one user directory is listed. If more than one user directory is listed, the sample application script asks the user to enter the user directory name in this file. The default value does not exist.

USER_ATTRIBUTE

Indicates that the value of this attribute becomes the Name ID value in the SAML assertion. If no value is specified for this setting, the sample application script chooses a value based on the user directory type. Example of attribute values can include:

- LDAP: uid or mail
- ODBC: name or email

If no value is specified, the following defaults are used:

- For LDAP: uid
- For ODBC: name
- For ActiveDirectory: cn

AGENT_NAME

Defines the name of the DefaultAgentName configuration setting for the Web Agent. This setting is specified in the Agent Configuration Object of the Policy Serve User Interface. If no value is specified for this setting, the sample application script reads the DefaultAgentName from the policy store, provided only one Agent configuration object found in the policy store. If more than one Agent configuration object exists, the sample application prompts the user to enter the DefaultAgentName value in this file.

WEB_SERVER_DOC_ROOT

Specifies the full path to the document root directory of the web server. The default value is C:\Inetpub\wwwroot, the root directory for an IIS web server. For example, if you are using a Sun Java System web server, the path would be *server_root/docs*.

WEB_SERVER_PORT

Specifies the listening port of the web server. The default port is 80.

PARTNER_WEB_SERVER_PORT

Specifies the listening port of the web server on the opposite side of the federation connection. For example, if your site is the IdP, then this site is the SP web server port. The default port is 80.

Modify the SetupFederationSample.pl Script (Optional)

The SetupFederationSample.pl script executes the sample application. This script resides in the directory *policy_server_home/samples/federation*.

The SetupFederationSample.pl script deploys the sample application. The script accomplishes these tasks:

- Reads the configuration information from the FederationSample.conf file.
- Creates policy objects in the policy store to establish the SAML 2.0 single sign-on and single logout profiles.
- Copies web pages to the web server document root
- Adds a private key/certificate pair to the certificate data store.
- Modifies the hosts file of the system to map a loopback IP address, 127.0.0.1 to www.sp.demo and www.idp.demo.

Important! If you install the Policy Server and the Web Agent Option Pack on different machines, comment out the call to CheckPreRequisites() in the SetupFederationSample.pl file.

To comment out the CheckPreRequisites() call

1. Open the script in an editor.
2. Comment out the CheckPreRequisites line, as shown here:

```
if ($CURRENT_COMP == $COMP_FSS)
{
#   CheckPreRequisites();
}
```
3. Save the script.

SetupFederationSample.pl Script Options (fss)

The SetupFederationSample.pl script uses the following command options:

-admin

Specifies the user name of the SiteMinder Administrator.

-password

Specifies the password of the SiteMinder Administrator in clear text.

-remove

Removes all objects that the sample application creates.

-idp

Creates only the Identity Provider objects in the policy store. You cannot use this option and the -sp option together. If you do not specify a value for this option or the -sp option, the sample application assumes a default of SiteMinder-to-SiteMinder communication.

Options: FSS, SMFE

-sp

Creates only Service Provider policy objects in the policy store. You cannot use this option and the -idp option together.

Options: FSS, SMFE

-partner

(optional) Indicates which application is installed at the partner site. The default is FSS.

Options: FSS, SMFE

Important! All the command line options are case-sensitive.

Run the Sample Application on the Policy Server System

Deploy the sample application on the Policy Server system.

You must have read/write permissions to the document root directory of the web server to run the sample application script.

Note: Run the SetupFederationSample.pl script once. If you run it again, the script deletes the sample policy objects that the previous execution of the script created.

Before you run the sample application:

1. Complete all [prerequisites](#) (see page 84).
2. Modify the FederationSample.conf file.
3. (Optional) Modify the [SetupFederationSample.pl script](#) (see page 88).

To run the sample application

1. Open a command window.
2. Navigate to *policy_server_home/siteminder/samples/federation*.

3. Run the SetupFederationSample.pl script using the Perl interpreter that is shipped with SiteMinder. This script is in the directory *policy_server_home/CLI/bin*.

```
perl SetupFederationSample.pl -admin siteminder_administrator  
-password administrator_password
```

Example:

```
perl SetupFederationSample.pl -admin siteminder -password mypassword
```

Important! All the command line options are case-sensitive.

4. Enter **yes** when you are prompted to continue with the installation. Do not enter the letter "y."

You can review the list of [script command options](#) (see page 88).

Using Multiple Policy Servers for the Sample Application

To establish a physically distinct Identity Provider and a Service Provider, you can set up a four-system environment.

The Identity Provider site uses a Policy Server and a Web Agent with the Web Agent Option Pack. The Service Provider site uses a second Policy Server and a Web Agent with the Web Agent Option. The Policy Servers and Web Agents with Option Packs are on separate systems.

If you set up a four-system environment, run the SetupFederationSample.pl script on both Policy Server systems. Use one of the following commands:

- On the IdP Policy Server, enter:

```
perl SetupFederationSample.pl -admin siteminder_administrator  
-password administrator_password -idp FSS
```

- On the SP Policy Server, enter:

```
perl SetupFederationSample.pl -admin siteminder_administrator  
-password administrator_password -sp FSS
```

You can review the list of [script command options](#) (see page 88).

Set up the Web Agent System

Set up the Web Agent system to use the sample application.

Follow these steps:

1. On the Policy Server system, navigate to the web server root directory that you specified in the FederationSample.conf file. Refer to the WEB_SERVER_DOC_ROOT setting.
2. Copy the idpsample and spsample directory *web_agent_home/affwebservices* on the Web Agent system.

3. Add a virtual directory mapping for the idpsample and spsample directories. Map to the following physical directories:

web_agent_home/affwebservices/idpsample

web_agent_home/affwebservices/spsample

4. Add mappings for `www.idp.demo` and `www.sp.demo` to the hosts file for the Web Agent system.

Windows

The host file is typically in `WINDOWS\system32\drivers\etc\hosts`.

UNIX

The host file is commonly in `/etc/hosts`.

Note: You can access the sample application through a browser on any system; however, the system must have the correct host mappings for `www.idp.demo` and `www.sp.demo`.

Using Multiple Web Agents for the Sample Application

To establish a physically distinct Identity Provider and a Service Provider, you can set up a four-system environment.

The Identity Provider site uses a Policy Server and a Web Agent with the Web Agent Option Pack. The Service Provider site uses a second Policy Server and a Web Agent with the Web Agent Option. The Policy Servers and Web Agents with Option Packs are on separate systems.

If you set up a four-system environment, modify the host file of each Web Agent system. The Web Agent must be able to recognize the other system with which it is communicating.

- At the Identity Provider Web Agent, modify the host file of this system to include the IP address of the SP system, `www.sp.demo`.
- At the Service Provider Web Agent, modify the host file of this system to include the IP address of the Identity Provider system, `www.idp.demo`.

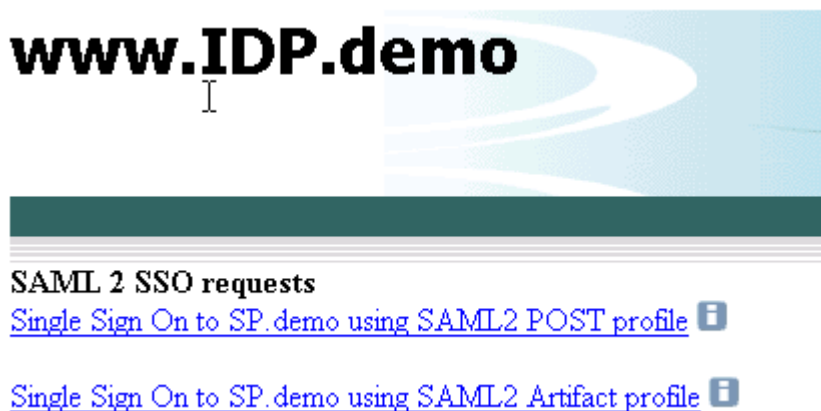
Test Single Sign-on with the Sample Application

After you run the sample application, test single sign-on.

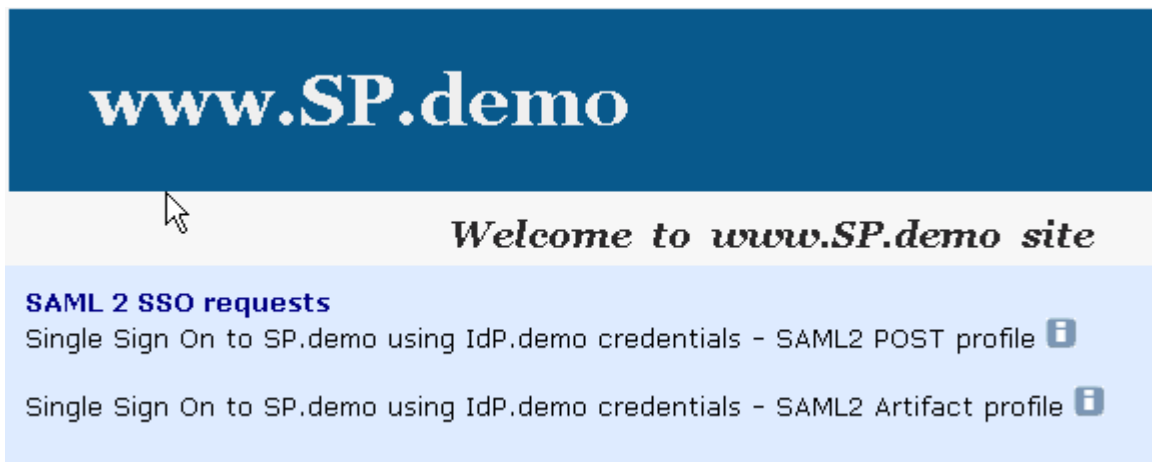
To test federated single sign-on

1. Open up a browser.
2. Enter the URL for the web page that has links to trigger single sign-on.
 - For IdP-initiated single sign-on, access the index.jsp page at:
`http://www.idp.demo:server_port/idpsample/index.jsp`
 - For SP-initiated single sign-on, access the index.jsp page at:
`http://www.sp.demo:server_port/spsample/index.jsp`

The following illustration is the IdP.demo home page:

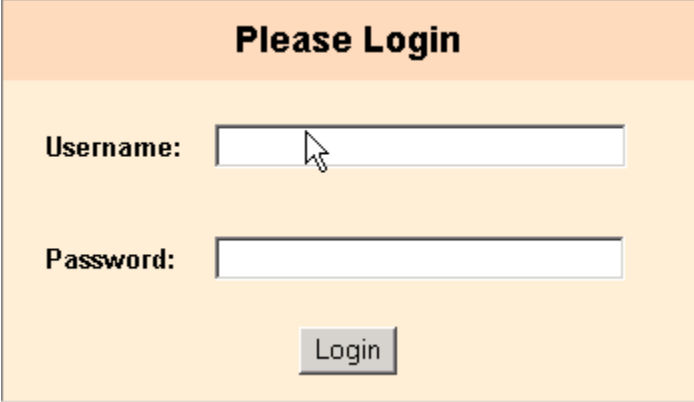


The following illustration is the SP.demo home page:



3. Click on one of the single sign-on links.

A login challenge like the following dialog is presented:



The image shows a login dialog box with an orange header containing the text "Please Login". Below the header, there are two input fields: "Username:" followed by a text box with a mouse cursor, and "Password:" followed by a text box. At the bottom center of the dialog is a "Login" button.

4. Using the login of an existing user in your user store, enter the user credentials. For example, if user1 is a user in the user store, enter the credentials for this user.

If single sign-on is successful, the following welcome page appears:



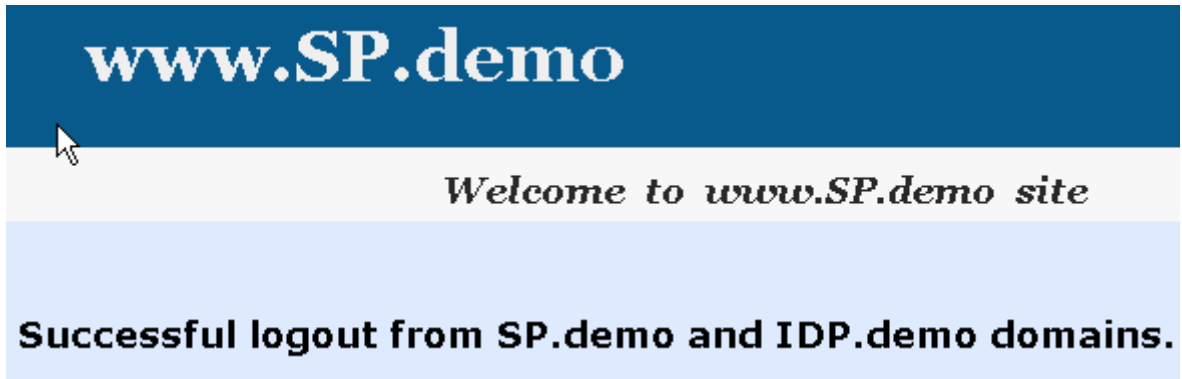
Test Single Logout with the Sample Application

After you have successfully tested single sign-on, you can test single logout from the SP.demo welcome page.

To test single logout

On the SP Welcome page, click the link labeled Single Logout using HTTP Redirect binding.

If single logout is successful, the following page appears:



Review Application-Generated SiteMinder Objects

The sample application automatically creates policy objects that enable the federated single sign-on and logout processes. After you successfully sign on, log on to the FSS Administrative UI and look at the various Policy Server objects set up by the sample application.

Objects to look at include:

- sp.demo in the IdP Federation Sample Partners affiliate domain
The sample application creates this Service Provider Properties object.
- Partner Idp.demo Auth Scheme
The sample application creates this SAML Authentication Scheme for the SP-site.

To see the SAML assertion generated by SiteMinder, look at the FWSTrace.log, located in the directory *web_agent_home/log*.

Note: Enable trace logging in the LoggerConfig.properties file to create a trace log. The LoggerConfig.properties file is located in *web_agent_home/affwebservices/WEB-INF/classes*.

Chapter 3: Deploy Federation Using a Manual Configuration

This section contains the following topics:

[Manual SiteMinder-to-SiteMinder Deployment Overview](#) (see page 95)

[Manual Deployment Prerequisites](#) (see page 96)

[Sample Federation Network](#) (see page 96)

[Set Up the Identity Provider](#) (see page 102)

[Set Up the Service Provider](#) (see page 118)

[Test SAML 2.0 Single Sign-on](#) (see page 131)

[Add Functionality to the Federation Deployment](#) (see page 134)

Manual SiteMinder-to-SiteMinder Deployment Overview

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

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

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

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

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

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

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

Manual Deployment Prerequisites

This deployment assumes that you know how to do the following:

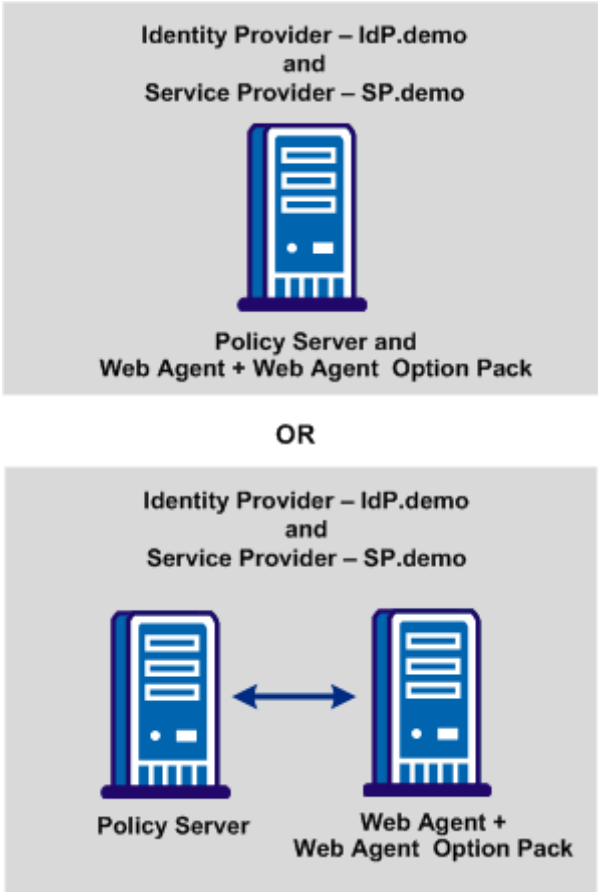
- Install and configure the SiteMinder Policy Server and Web Agent and any associated Option Pack.
- Enable a Web or application server for SSL communication (needed for artifact binding).
- Work with certificates and understand certificate operation, such as how to request a certificate and have it signed by a certificate authority, know the difference between a private key and a public key.
- Add users to a user store. For example, if you have a Sun ONE Directory Server, you must know how to use the Sun ONE Server Console.
- Set up an ODBC database to be enabled as a session store.

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

Sample Federation Network

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

The following illustration shows the sample federated network.



Identity Provider Data for a Basic Configuration

IdP.demo is the Identity Provider. The following two tables list the set-up for a basic SAML 2.0 POST configuration at the site and then a more advanced configuration. You can also fill in information for your network.

The following table contains sample data required for the most basic SAML 2.0 POST single sign-on configuration.

Identity Provider Component	Sample Network	Your Network
IdP Policy Server	Server: www.idp.demo:80 Server type: IIS 6.0 Web Server	Server: Server type:
IdP policy store	IP Address: www.idp.demo:389 Storage: LDAP (Sun One Directory Server) Root DN: o=idp.demo Admin Username: cn=Directory Manager Password: federation	IP Address: Storage: Root DN: Admin Username: Password:
User store	Server: www.idp.demo:42088 Server Type: Sun One Directory Server (LDAP) User store: The LDAP directory contains the following users: <ul style="list-style-type: none"> ■ Tuser1 ■ Tuser2 userpassword: test mail: <user_name>@idp.demo Root: dc=idp,dc=demo Start: uid= End: ,ou=People,dc=idp,dc=demo	Server: Server Type: User store: Users passwords: Attribute: Attribute description: Root: Start: End:
IdP Web Agent with Web Agent Option Pack	Server: www.idp.demo:80 Server Type: IIS 6.0 Web Server Agent name: idp-webagent	Server: Server Type: Agent name:

Identity Provider Component	Sample Network	Your Network
Assertion Consumer Service URL	URL: http://www.sp.demo:81/affwebservice/public/saml2assertionconsumer	URL:
Assertion Retrieval Service URL	URL: http://www.idp.demo:80/affwebservice/assertionretriever	
Authentication URL	URL: http://www.idp.demo/siteminderagent/redirectjsp/redirect.jsp	URL:

Identity Provider Data for an Advanced Configuration

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

Identity Provider Component	Sample Network	Your Network
Session server	Server: www.idp.demo Database type: ODBC Database Source Information: SiteMinder Session Data Source User Name: admin Password: dbpassword	Server: Database type: Database Source Information: User Name: Password:
SSL-enabled server	Server: www.idp.demo:443 Server Type: IIS 6.0 Web The web server with the Web Agent Option Pack is SSL-enabled for artifact binding	Server: Server Type:
Certificate of the Certificate Authority (CA)	Certificate of CA: docCA.crt DER-encoded Cert: docCA.der This CA signs the server-side certificate to enable SSL	Certificate of CA: DER-encoded Cert:

Identity Provider Component	Sample Network	Your Network
Private key/certificate pair to sign SAML responses	Certificate: post-cert.crt Private key: post-pkey.der Password: fedsvcs	Certificate: Private key: Password:
Certificate (public key) for encryption	Public key: sp-encrypt.crt	Public key:
Attribute to include in assertion	Attribute: unspecified (default) Attribute Kind: User DN Variable Name: firstname Variable Value: givenname	Attribute: Attribute Kind: Variable Name: Variable Value:

Service Provider Data for a Basic Configuration

Sp.demo is the Service Provider. The following two tables list the set-up for a basic SAML 2.0 POST configuration of the site and a more advanced SAML 2.0 configuration. You can also fill in information for your network.

The following table contains sample data required for the most basic SAML 2.0 POST single sign-on configuration.

Service Provider Component	Sample Network	Your Network
SP Policy Server	Server: www.sp.demo:80 Server type: IIS 6.0 Web Server	Server: Server type:
SP policy store	IP Address: www.sp.demo:389 Storage: LDAP (Sun One Directory Server) Root DN: o=ca.com Admin Username: cn=Directory Manager Password: federation	IP Address: Storage: Root DN: Admin Username: Password:

Service Provider Component	Sample Network	Your Network
User Store	Server: www.sp.demo:32941 Server Type: LDAP (Sun One Directory Server) User store: The LDAP directory contains the following users: <ul style="list-style-type: none"> ■ Tuser1 ■ Tuser2 userpassword: customer mail: <user_name>@sp.demo Root: dc=sp,dc=demo Start: uid= End: ,ou=People,dc=sp,dc=demo	Server: Server Type: User store: User passwords: Users password: Attribute: Attribute description: Root: Start: End:
SP Web Agent and Web Agent Option Pack	Server: www.sp.demo:81 Server type: Sun ONE 6.1 Web Server Agent name: sp-webagent	Server: Server type: Agent name:
Single Sign-on Service	SSO Service: http://www.idp.demo:80/affwebservices/public/saml2sso	SSO Service:
Target Resource	Target Resource: http://www.sp.demo:81/spsample/protected/target.jsp	Target:

Service Provider Data for an Advanced Configuration

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

Service Provider Component	Sample Network	Your Network
Artifact Resolution Service	Resolution Service: https://www.idp.demo:443/affwebservices/saml2artifactresolution	Resolution Service:

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

Set Up the Identity Provider

To deploy Federation Security Services at the Identity Provider, the following information details the tasks.

Install the IdP Policy Server

Set up the Policy Server.

To install the Policy Server

1. Install a Policy Server.

For instructions, see the *SiteMinder Policy Server Installation Guide*.

2. Select the web server that is used for the UI.

In this deployment, an IIS Web Server is the server on which the Policy Server is installed. Your network can use a different supported web server.

3. Select a policy store.

In this deployment, a Sun Java LDAP directory is serving as the policy store. The installation configures and initializes this policy store for you.

Important! If you initialize a new policy store, the Policy Server installer automatically imports the affiliate objects contained in the `ampolicy.smdif` file. These objects are necessary for federation. If you use an existing policy store that you do not initialize, import the affiliate objects manually. To verify that the import is successful, log in to the FSS Administrative UI and click on Domains in the System tab. If the import is successful, you can see the FederationWebServices domain object.

4. (Optional) Enable Policy Server Trace Logging so you can use the log to troubleshoot your setup.
5. Point the Policy Server to the LDAP Policy Store.

More information:

[Point the Policy Server to the IdP LDAP Policy Store](#) (see page 103)

[Enable Policy Server Trace Logging at the IdP](#) (see page 105)

Point the Policy Server to the IdP LDAP Policy Store

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

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

Follow these steps:

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

Databases

Policy Store

Storage

LDAP

IP Address (LDAP directory)

www.idp.demo:389

Root DN

o=idp.demo

Admin Username

cn=Directory Manager

Password

password

Confirm Password

password

4. Click OK to save your changes and exit the console.
5. Go to [Set Up the IdP User Store](#) (see page 104).

Set Up the IdP User Store

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

To configure the user store

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

user1

userpassword: test

mail: user1@idp.demo

user2

userpassword: test

mail: user2@idp.demo

Important! The email address must be the same in the Service Provider user store for the same users.

3. [Enable trace logging](#) (see page 105).

Enable Policy Server Trace Logging at the IdP

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

Follow these steps:

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

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

You configure trace logging at the Policy Server using the Policy Server Management Console.

3. [Install the IdP Web Agent](#) (see page 105).

Install the IdP Web Agent

Install the Web Agent at the IdP for the sample deployment.

Follow these steps:

1. Install a Web Agent on a supported web server.

For instructions on installing a Web Agent, see *SiteMinder Web Agent Installation Guide*.

In this deployment, the web server is an IIS Web Server. Your web server can be different.

2. Register the server with the Agent as a trusted host.
3. Enable the Web Agent in the `WebAgent.conf` file.

In this deployment, the Web Agent is installed on an IIS Web Server.

4. Install the IdP Web Agent Option Pack.

Install the IdP Web Agent Option Pack

The Web Agent Option pack installs the Federation Web Services (FWS) application. FWS is a required component for SiteMinder federation.

To set up the Web Agent Option Pack

1. Install the Web Agent Option Pack on the same web server as the Web Agent. In this deployment, the server is an IIS Web Server.

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

2. [Configure the Web Server with the Web Agent Option Pack](#) (see page 106).

Configure the Web Server with the Web Agent Option Pack

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

To set up FWS:

- [Install the JDK for Federation Web Services](#) (see page 106)
- [Install and Configure ServletExec to work with FWS at the IdP](#) (see page 106)
- [Configure the AffWebServices.properties File at the IdP](#) (see page 109)
- [Test Federation Web Services at the IdP](#) (see page 109)

Install the JDK for Federation Web Services

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

For the correct JDK version, go to the [Technical Support site](#) and search for the SiteMinder Platform Support Matrix for the release.

Install and Configure ServletExec to work with FWS at the IdP

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

Note: SiteMinder r12.0 SP3 is shipped with a ServletExec license key file named ServletExec_AS_6_license_key.txt. If you do not have this license key, contact [CA Technical Support](#). From this license file, copy the license key and enter it in the ServletExec License dialog of the ServletExec Administration Console. For instructions on licensing ServletExec, see ServletExec documentation, available at the New Atlanta Communication <http://www.newatlanta.com> website.

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

To set up ServletExec

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

The Manage Web Applications dialog opens.

4. Click Add a Web Application.
5. Enter the following information:

Application Name

affwebservices

URL Context Path

/affwebservices/

Location

C:\program files\ca\webagent\affwebservices

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

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

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

More Information:

[Enable ServletExec to Write to the IIS File System](#) (see page 107)

[Configure the FWS Properties File at the IdP](#) (see page 109)

Enable ServletExec to Write to the IIS File System

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

Follow these steps:

1. Open the IIS Internet Information Services Manager on the system where ServletExec is installed.
2. Navigate to Web Sites, Default Web Site.
The set of applications is displayed in the right pane.
3. Select ServletExec and right-click Properties.
4. Select the Directory Security tab in the Properties dialog.
5. Click Edit in the Authentication and access control section.
The Authentication Methods dialog opens.
6. Set the controls as follows.
 - a. Select Enable Anonymous Access.
For anonymous access, enter a name and password of a user account that has the permissions to right to the Windows file system. To grant this right to a user account, see Windows documentation. For example, you can use the IUSR Internet Guest account for anonymous access.
 - b. Clear Basic authentication.
 - c. Clear Integrated Windows authentication.
7. If prompted, apply the security changes to all child components of the web server.
8. Restart the web server.

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

Follow these steps:

1. Open Control Panel, Administrative Tools, Local Security Policy, Local Policies, User Rights Assignment.
The Local Security Settings dialog displays.
2. Double-click Act as part of the operating system.
The Act as part of the operating system Properties dialog opens.
3. Add the anonymous user account to the Local Security Setting dialog.
4. Click OK.
5. Exit from the control panel.
6. Optionally, we strongly recommend that you look at the Agent Configuration Object for the Web Agent protecting the IIS Web Server. This object verifies that the SetRemoteUser parameter is set to yes to preventing any anonymous user from writing to the file system.

Configure the FWS Properties File at the IdP

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

To modify the `affwebservices.properties` file

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

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

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

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

Verify that this path is entered on one line.

3. Save and close the file.
4. [Test Federation Web Services at the IdP](#) (see page 109).

Test Federation Web Services at the IdP

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

Follow these steps:

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

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

fqhn

Defines the fully qualified host name.

port_number

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

For this deployment, enter:

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

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

Assertion Retrieval Service has been successfully initialized.

The requested servlet accepts only HTTP POST requests.

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

2. [Enable Web Agent Option Pack Logging at the IdP](#) (see page 110).

Enable Web Agent Option Pack Logging at the IdP

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

- `affwebservices.log`
- `FWSTrace.log`

Follow these steps:

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

More Information:

[Federation Security Services Trace Logging](#) (see page 463)

Specify the User Store for the IdP Policy Server

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

The following steps specify how to configure a user directory in the FSS Administrative UI. The directory, named IdP LDAP, is the Sun ONE LDAP directory that contains the users `Tuser1` and `Tuser2`.

To configure a user directory

1. Log in to the FSS Administrative UI.
2. Select the System tab.
3. Select Edit, System Configuration, Create User Directory.
The User Directory Properties dialog opens.
4. Complete the following fields in the Directory Setup section:

Name

IDP LDAP

In the Directory Setup section:

NameSpace

LDAP

Server

www.idp.demo:42088

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

Root

dc=idp,dc=demo

Accept the defaults for the other values.

Complete the following field in the LDAP User DN Lookup section:

Start

uid=

End

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

6. Click View Contents to verify you can view the contents of the directory.
7. Click Submit.
8. [Set up an Affiliate Domain at the IdP](#) (see page 112).

Set up an Affiliate Domain at the IdP

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

To configure an affiliate domain

1. Log in to the FSS Administrative UI.
2. From the menu bar of the SiteMinder Administration window, select Edit, System Configuration, Create Domain.

The Domain Properties dialog opens.

3. Complete the following fields:

Name

Federation Sample Partners

Description

Domain for sp.demo

4. In the Domain Type section, select Affiliate Domain.
5. Leave this dialog open and [Add the User Directory to the Affiliate Domain at the IdP](#) (see page 112).

Add the User Directory to the Affiliate Domain at the IdP

To associate a user directory for the affiliate domain

1. Navigate to the Users setting.

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

Select the IDP LDAP directory.

2. Click Add Members.

The Users/Groups page displays.

3. Complete the fields to select users from the user directory.
4. Click OK.
5. Go to [Add the Service Provider to the Affiliate Domain at the IdP](#) (see page 113).

Add the Service Provider to the Affiliate Domain at the IdP

To add sp.demo to the affiliate domain, specify values on the Users tab, the General tab, and the SSO tab before you can save a Service Provider object.

To add sp.demo to the Federation Sample Partners domain

1. Begin at the Domains tab.
2. Select Federation Sample Partners, right-click, and select Create SAML Service Provider.
3. Complete the following fields:

Name

sp.demo

Description

Service Provider

Authentication URL

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

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

After successful authentication, the redirect.jsp application redirects the user back to the SSO service for assertion generation. A SiteMinder policy must [protect this URL](#) (see page 113).

Enabled

Verify that this option is selected. By default, this option is selected.

4. Keep the Policy Server User Interface open and [Select Users For Which Assertions Will Be Generated at the IdP](#) (see page 115).

Protect the Authentication URL (SAML 2.0)

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

To protect the Authentication URL at the Identity Provider

1. From the Domains tab, create a policy domain called Authentication URL Protection Domain.
2. Add the IdP LDAP user directory in the User Directories tab.

3. From the Authentication URL Protection domain, create a persistent realm with the following field entries:

Name

Authentication URL Protection Realm

Agent

Using the lookup button, select FSS web agent

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

Resource Filter

/siteminderagent/redirectjsp/redirect.jsp

Accept the defaults for the other settings.

Session tab

Select Persistent Session

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

Name

Authentication URL Protection Rule

Realm

Authentication URL Protection Realm

Resource

*

Web Agent actions

Get

Accept the defaults for the other settings.

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

Name

Authentication URL Protection Policy

Users tab

Add user1 from the IdP LDAP user directory

Rules tab

add Authentication URL Protection Rule

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

Select Users for which the IdP Generates Assertions

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

To select users that use assertions as credentials

1. Log in to the FSS Administrative UI.
2. From the Domains tab, expand Federation Sample Partners and select SAML Service Providers to display the Service Providers.
3. Select sp.demo and right-click to open the properties of this Service Provider.
4. From the Users tab of the SAML Service Provider Properties dialog, select the IdP user store tab. In this deployment, select the IdP LDAP tab.
5. Click Add/Remove.

The Users/Groups dialog opens.

6. Search the Available Members list for Tuser1 and Tuser2. These employees are listed in the IdP LDAP directory.
 - a. Click the binoculars icon under the Available Members list.
 - b. In the Search LDAP/AD Directory dialog, select Attribute-Value Pair and complete the fields as follows:

Attribute

uid

Value

*

- c. Click OK. The individual users in the IdP LDAP directory are displayed.
 - d. Hold the CTRL or SHIFT key, and select the entries for Tuser1 and Tuser2. Then, click the left arrow to move them to the Current Members list.
7. Click OK to return to the SAML Service Providers Properties dialog.
 8. [Configure a Name ID for Inclusion in the Assertion](#) (see page 116).

Configure a Name ID for Inclusion in the Assertion

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

To configure name IDs

1. Select the Name IDs tab on the SAML Service Provider Properties dialog.
2. Complete the following fields:

Name ID Format

Unspecified

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

Name ID Type section

User Attribute

Attribute Name

mail

3. Keep the SAML Service Provider Properties dialog open and [Identify the SP, IdP, and Other General Settings](#) (see page 116).

Identify the SP, IdP, and Other General Settings

Identify the Service Provider and the Identity Provider. The IdP ID identifies the Issuer of the assertion. The SP ID is used to accept the AuthnRequest when it is sent from the Service Provider.

To configure general settings

1. Select the General tab on the SAML Service Providers dialog box.

Configure the following fields:

SP ID

sp.demo

IdP ID

idp.demo

The values for the SP ID and IdP ID must match the values at the Service Provider.

SAML Version

2.0 (default)

Skew Time

30 seconds (default)

2. In the D-Sign Info box, select the Disable Signature Processing checkbox.

Important! Disabling signing is intended *only* for debugging the initial single sign-on configuration. In a production environment, enable signature processing, which is a mandatory security requirement.

3. Keep the SAML Service Provider Properties dialog open and [Configure POST Single Sign-on at the IdP](#) (see page 117).

More Information:

[Configure Digital Signing \(required for POST Binding\)](#) (see page 145)

Configure POST Single Sign-on at the IdP

You need to specify the SAML 2.0 binding you want to use for single sign-on.

To configure single sign-on with POST binding

1. Select the SSO tab.

Complete the following fields:

Audience

sp.demo

Assertion Consumer Service

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

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

HTTP-POST

select this check box

Authentication Level

5 (default)

Validity Duration

60 (default)

In a test environment, if you see the following message in the Policy Server trace log,

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

you may want to increase the Validity Duration value above 60.

AuthnContext Class Ref

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

2. Accept the default values for all other remaining fields.
3. Click OK.
4. [Protect the Authentication URL](#) (see page 113).

Federation Web Services Access

A Web Agent protects the FWS application. The Agent must be bound to the realms that contain the servlets that make up the FWS application. To associate the Web Agent with the realms, add the Agent to the default Web Agent group. The default Web Agent group is established as part of the Web Agent Option Pack installation. Additionally, the Service Providers need access to the Assertion Retrieval Service to obtain assertions.

To allow access to Federation Web Services

Add the Web Agent that protects Federation Web Services to the Agent group FederationWebServicesAgentGroup.

This associates the Agent with the default realms.

Configure the Service Provider

After completing the configuration at the Identity Provider, you must [Set Up the Service Provider](#) (see page 118).

Set Up the Service Provider

A number of steps are involved in setting up the Service Provider in a federation network.

Install the SP Policy Server

At the Service Provider, install the Policy Server.

Set up the Policy Server.

To install the Policy Server

1. Install a Policy Server.

For instructions, see the *SiteMinder Policy Server Installation Guide*.

2. Select the web server that is used for the UI.

In this deployment, an IIS Web Server is the server on which the Policy Server is installed. Your network can use a different supported web server.

3. Select a policy store.

In this deployment, a Sun Java LDAP directory is serving as the policy store. The installation configures and initializes this policy store for you.

Important! If you initialize a new policy store, the Policy Server installer automatically imports the affiliate objects contained in the `ampolicy.smdif` file. These objects are necessary for federation. If you use an existing policy store that you do not initialize, import the affiliate objects manually. To verify that the import is successful, log in to the FSS Administrative UI and click on Domains in the System tab. If the import is successful, you can see the FederationWebServices domain object.

4. (Optional) Enable Policy Server Trace Logging so you can use the log to troubleshoot your setup.

Point the Policy Server to the LDAP Policy Store.

More information:

[Enable Trace Logging for Federation Components at the SP](#) (see page 121)

[Point the Policy Server to the SP LDAP Policy Store](#) (see page 120)

Point the Policy Server to the SP LDAP Policy Store

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

Follow these steps:

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

Complete the following fields:

Databases

Policy Store

Storage

LDAP

LDAP IP Address

sp.demo:389

Root DN

o=sp.demo

Admin Username

cn=Directory Manager

Password

federation

Confirm Password

federation

3. Click OK.
4. [Set up the SP user store](#) (see page 120).

Set Up the SP User Store

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

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

To configure the user store

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

user1

userpassword: customer

mail: user1@sp.demo

user2

userpassword: customer

mail: user2@sp.demo

Important! The email address must be the same in the Identity Provider user store for the same users.

3. [Enable trace logging](#) (see page 121).

Enable Trace Logging for Federation Components at the SP

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

To enable logging

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

To configure trace logging at the Policy Server, using the Policy Server Management Console.
3. [Install the SP Web Agent](#) (see page 122).

Install the SP Web Agent

Install the Web Agent at the Service Provider

Follow these steps:

1. Install a Web Agent on a supported web server.

For instructions on installing a Web Agent, see *SiteMinder Web Agent Installation Guide*.

In this deployment, the server is a Sun ONE 6.1 Web Server. Your server can be different.

2. Register the system with the Agent as a trusted host.
3. Enable the Web Agent in the WebAgent.conf file.
4. Install the SP Web Agent Option Pack.

Install the SP Web Agent Option Pack

The Web Agent Option pack installs the Federation Web Services (FWS) application.

To set up the Web Agent Option Pack

1. Install a JDK.

For the supported version of the JDK, see the SiteMinder r12 Platform Support Matrix on the [Technical Support site](#). This matrix includes r12.0 SP3.

Install the Web Agent Option Pack on the same web server as the Web Agent.

In this deployment, the server is an IIS Web Server.

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

2. [Configure the Web Server with the Web Agent Option Pack](#) (see page 122).

Configure the Web Server with the Web Agent Option Pack

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

For FWS to work, do the following

1. [Install the JDK for Federation Web Services](#) (see page 123)
2. [Install and Configure ServletExec to Work with FWS at the SP](#) (see page 123)

3. [Configure the AffWebServices.properties file](#) (see page 124)
4. [Enable Web Agent Option Pack logging](#) (see page 125)
5. [Test Federation Web Services](#) (see page 125)

Install the JDK for Federation Web Services

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

Install and Configure ServletExec to Work with FWS at the SP

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

Note: SiteMinder r12.0 SP3 is shipped with a ServletExec license key file named ServletExec_AS_6_license_key.txt. If you do not have this license key, contact [CA Technical Support](#). From this license file, copy the license key and enter it in the ServletExec License dialog of the ServletExec Administration Console. For instructions on licensing ServletExec, see ServletExec documentation, available at the New Atlanta Communication <http://www.newatlanta.com> website.

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

To set up ServletExec

1. Install ServletExec.

For instructions, refer to New Atlanta Communications documentation.

2. Open the ServletExec Administration Console.
3. Under Web Applications, select manage.

The Manage Web Applications dialog opens.

4. Click Add a Web Application.
5. Enter the following information:

Application Name

affwebservices

URL Context Path

/affwebservices/

Location

C:\program files\ca\webagent\affwebservices

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

6. Click Submit.
7. Exit the ServletExec Console.
8. [Configure the AffWebServices.properties file](#) (see page 124).

Configure the FWS Properties File

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

Follow these steps:

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

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

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

Note: Federation Web Services is a Java component, so the Windows paths must contain double backslashes. Specify this entry on one line.

3. Save and close the file.
4. [Test Federation Web Services](#) (see page 125).

Test Federation Web Services

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

Follow these steps:

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

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

fqhn

Defines the fully qualified host name.

port_number

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

For this deployment, enter:

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

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

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

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

2. [Enable Web Agent Option Pack logging.](#) (see page 125)

Enable Web Agent Option Pack Logging at the SP

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

- `affwebserv.log`
Contains error logging messages.
- `FWSTrace.log`

To enable error and trace logging

1. Open up the `LoggerConfig.properties` file. This file can be found in the directory `web_agent_home/affwebservices/WEB-INF/classes`.
2. Set the `LoggingOn` parameter to `Y`.

3. Accept the default name and location for the LogFileName setting, which points to the affwebserv.log file.
4. Set the TracingOn setting to Y.
5. Accept the default name and location for the TraceFileName setting, which points to the FWSTrace.log file.

Logging is now enabled.

More Information:

[Federation Security Services Trace Logging](#) (see page 463)

Specify the User Store for the SP Policy Server

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

The following steps specify how to configure a user directory in the FSS Administrative UI. The directory, named SP LDAP, is the Sun ONE LDAP directory that contains the users Tuser1 and Tuser2.

To configure a user directory

1. Log in to the FSS Administrative UI.
2. Select the System tab.
3. Select Edit, System Configuration, Create User Directory.

The User Directory Properties dialog opens.

4. Complete the following field:
Name
SP LDAP
5. Complete the following fields in the Directory Setup section:
Namespace
LDAP
Server
www.sp.demo:32941
6. Complete the following fields in the LDAP Search section:
Root
dc=sp,dc=demo
Accept the defaults for the other values.
7. Complete the following fields in the LDAP User DN Lookup section:
Start
uid=
End
,ou=People,dc=sp,dc=demo
8. Click View Contents to verify that you can view the contents of the directory.
9. Click Submit.

Specify the POST Binding Authentication at the SP

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

To select a single sign-on binding at the SP

1. Select the SSO tab from the SAML 2.0 Auth Scheme Properties dialog.
2. Complete the following fields:

Redirect Mode

302 Cookie Data (default)

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

SSO Service

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

Audience

sp.demo

This value must match the value at the Identity Provider.

Target

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

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

3. Select the HTTP-POST.
4. Clear the Enforce Single Use Policy option.
Disabling this option makes the sample network noncompliant with SAML 2.0. If you want to enable the use of the single use policy feature, set up a session store at the Service Provider.
5. Click OK until you exit the authentication scheme dialog.
6. Keep the Policy Server User Interface open and [Protect the Target Resource Using SAML 2.0 Authentication](#) (see page 129).

Configure the SAML 2.0 Authentication Scheme at the SP

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

To configure the SAML 2.0 authentication scheme

1. Log in to the FSS Administrative UI.
2. From the menu bar, select Edit, System Configuration, Create Authentication Scheme.

The Authentication Scheme Properties dialog opens.

3. Complete the following fields:

Scheme Common Setup section:

Name

Partner IDP.demo Auth Scheme

Authentication Scheme Type

SAML 2.0 Template

Protection Level

5 (default)

Scheme Setup tab fields:

SP ID

sp.demo

IdP ID

idp.demo

SAML Version

2.0 (default)

Skew Time

30 (default)

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

4. In the D-Sign Info box, select the Disable Signature Processing checkbox.

Important! Disabling signing is intended *only* for debugging the initial single sign-on configuration. In a production environment, signature processing is a mandatory security requirement. So signature validation must be enabled and the key store must be set up to validate signatures.

5. Click Additional Configuration.

The SAML 2.0 Auth. Scheme Properties dialog opens.

6. Leave the Authentication Scheme Properties dialog open and Configure User Disambiguation at the SP.

More information:

[Set Up smkeydatabase at the SP for Signature Validation](#) (see page 146)

Protect the Target Resource at the SP

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

To protect the target resource

1. From the System tab of the FSS Administrative UI, create a policy domain named Domain for IdP.demo Visitors.
2. Define a Web Agent. In this deployment, the Agent is sp-webagent. This Agent protects the server with the Web Agent Option Pack installed.
3. Associate the sp-webagent with the Domain for Idp.demo Visitors to protect the realm in this domain.
4. Add the user directory that holds users user1.

5. To the policy domain, add a persistent realm with the following components then click OK to save it.

Name

SP Target Page Protection Realm

Agent

sp-webagent

Resource Filter

Defines the path to the target resource at the Service Provider web server. For this deployment, the resource filter is /spsample/protected.jsp

Authentication Scheme

Partner IdP.demo Auth Scheme

Default Resource Protection

Protected

6. To the realm, add a rule with the following components then click OK to save it.

Name

SP Target Page Protection Rule

Realm

SP Target Page Protection Realm

Resource

*

Web Agent Actions

Get

Accept the defaults for all other fields.

7. Add a policy with the following components then click OK to save it.

Name

SP Target Page Protection Policy

Users

Add user1 so this user has access to the target

Rules

Add the SP Target Page Protection Rule

SiteMinder protects the target resource.

8. Exit the Policy Server User Interface.
9. Use HTML Pages to Test the Federation Set-up.

The protection policy for the target resource is complete.

Test SAML 2.0 Single Sign-on

To test single sign-on in a SiteMinder-to-SiteMinder network, use the web pages included with the sample application. You must have previously run the sample application script to access the web pages. If you do not run the sample application, use your own web pages to test single sign-on.

The sample application web pages are located in the following two folders.

`policy_server_home/samples/federation/content/idpsample`

`policy_server_home/samples/federation/content/spsample`

policy_server_home

Specifies the installed location of the SiteMinder Policy Server.

Important! If you have run the sample application, the `idpsample` and `spsample` folders are automatically copied into the document root directory of your web server.

If you use your own HTML page to test SP-initiated single sign-on, the HTML page must contain a hard-coded link to the AuthnRequest service. For this deployment, the sample link for POST binding is:

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

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

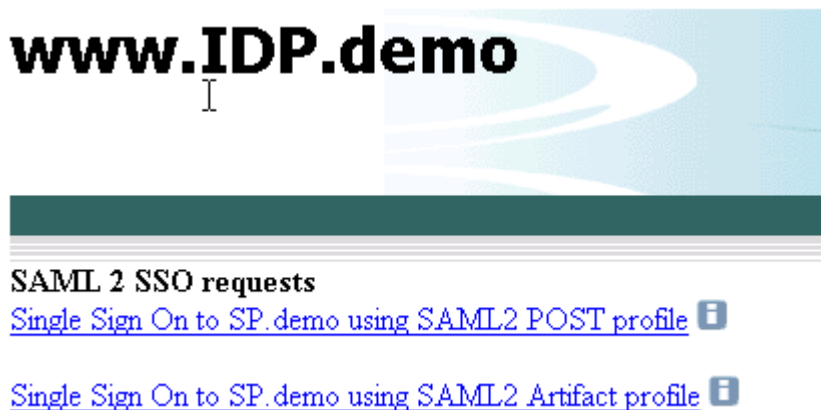
Note: The ProviderID in the Authnrequest link must match the IdP ID field value specified in the SAML authentication scheme at the SP. The IdP ID field is on the Scheme Setup tab of the Authentication Scheme Properties dialog.

After you run the sample application, test single sign-on.

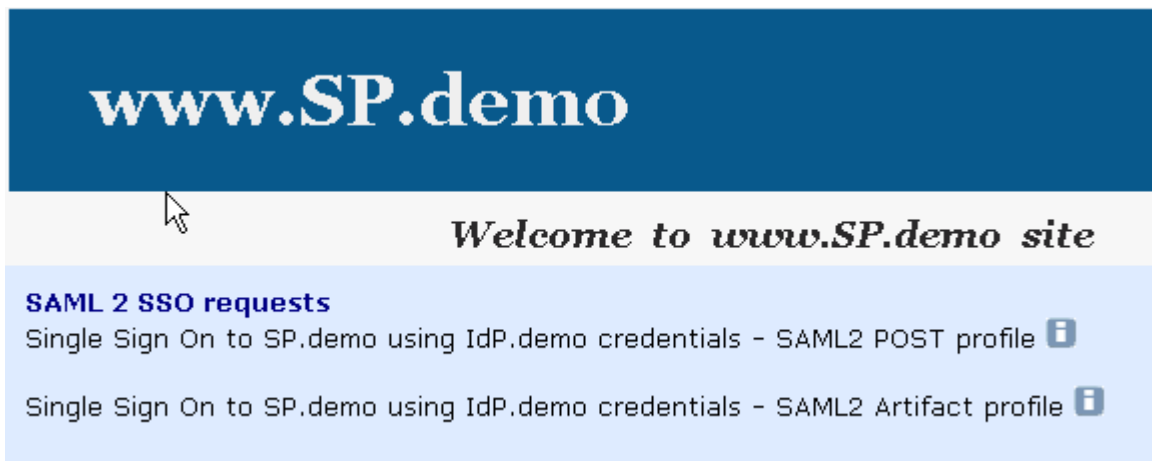
To test federated single sign-on

1. Open up a browser.
2. Enter the URL for the web page that has links to trigger single sign-on.
 - For IdP-initiated single sign-on, access the index.jsp page at:
`http://www.idp.demo:server_port/idpsample/index.jsp`
 - For SP-initiated single sign-on, access the index.jsp page at:
`http://www.sp.demo:server_port/spsample/index.jsp`

The following figure is the IdP.demo home page:

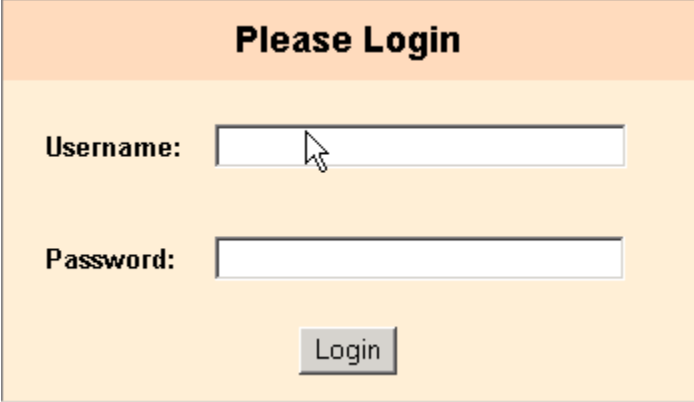


The following illustration is the SP.demo home page:



- Click on the SAML2 POST profile link.

The following login challenge appears:

A login form titled "Please Login" with a light orange background. It contains two input fields: "Username:" and "Password:". Below the fields is a "Login" button. A mouse cursor is positioned over the Username input field.

Please Login

Username:

Password:

Login

- Using the login of an existing user in your user store, enter the user credentials. For example, if user1 is a user in the user store, enter the credentials for this user.

If single sign-on is successful, the following welcome page appears:

A welcome page for the website www.SP.demo. The page has a dark blue header with the site name. Below the header is a light blue section with a welcome message and a SAML 2 SLO request notification.

www.SP.demo

Welcome to www.SP.demo site

Welcome Tuser1

SAML 2 SLO request
Single Logout using HTTP Redirect binding ⓘ

- After you test single sign-on, you can [Add Functionality to the Federation Deployment](#) (see page 134).

Add Functionality to the Federation Deployment

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

The additional tasks covered in this deployment example are:

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

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

Configure Single Logout

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

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

Enable Single Logout at the IdP

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

To configure single logout

1. Log in to the FSS Administrative UI and access the SAML Service Provider Properties dialog for `sp.demo`.
2. Select the SLO tab.
3. Select the HTTP-Redirect.

The remaining fields become active.

4. Enter values for the following fields:

SLO Location URL

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

Defines the SLO servlet at the SP.

SLO Confirm URL

`http://www.idp.demo:80/idpsample/SLOConfirm.jsp`

5. Accept defaults for the other fields.
6. From the Policy Server Management Console, enable the session server.

Enable Single Logout at the SP

You can initiate single logout at the Service Provider.

To configure single logout at the SP

1. Verify that the realm with the protected resources is configured for persistent sessions.
2. From the Authentication Scheme Properties dialog, click Additional Configuration.
The SAML 2.0 Auth Scheme Properties dialog opens.
3. Select the SLO tab.
4. Select the HTTP-Redirect checkbox.
The rest of the fields become active.
5. Complete the fields as follows:

SLO Location URL

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

SLO Confirm URL

`http://www.sp.demo:81/spsample/SLOConfirm.jsp`

6. Accept the default values for all other fields.
7. From the Policy Server Management Console, enable the session server.

Test Single Logout

Use the web pages included with the sample application to test single logout. To have access to these pages, you must have run the sample application.

The web pages are located in the following two folders.

`policy_server_home/samples/federation/content/idpsample`

`policy_server_home/samples/federation/content/spsample`

policy_server_home

Specifies the installed location of the SiteMinder Policy Server.

Important! If you have run the sample application, the `idpsample` and `spsample` folders are automatically copied into the document root directory of your web server.

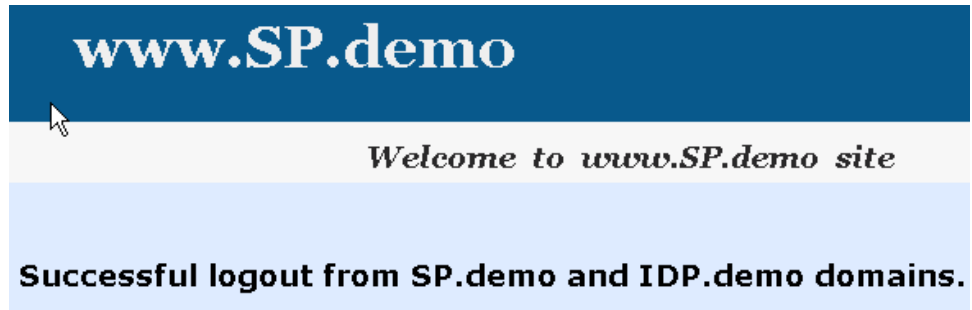
If you have not run the sample application, use your own web pages. Verify that your HTML page for testing SP-initiated single sign-on includes a hard-coded link to the single logout service.

After you have successfully tested single sign-on, you can test single logout from the SP.demo welcome page.

To test single logout

On the SP Welcome page, click the link labeled Single Logout using HTTP Redirect binding.

If single logout is successful, the following page appears:



Configure SAML 2.0 Artifact Single Sign-on

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

Required tasks at the Identity Provider:

- [Set up the IdP session store](#) (see page 137)
- [Enable SSL for the IdP web server](#) (see page 138)
- Permit access to the artifact resolution service policy
- Enable a persistent session to store assertions at the IdP
- Select the artifact binding at the IdP

Required tasks at the Service Provider:

- Add a CA certificate to the certificate data store at the SP
- Enable the artifact binding at the SP
- [Test artifact single sign-on](#) (see page 143)

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

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

To enable the session store

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

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

2. Open the Policy Server Management Console.
3. Select the Data tab.
4. Select Session Server From the Database drop-down list.
5. Complete the following fields:

Data Source Information

SiteMinder Session Data Source

User Name

admin

Password

dbpassword

Confirm Password

dbpassword

Maximum connections

16 (default)

6. Select the Enable Session Server check box.
7. Click OK to save the settings.
8. [Enable SSL for the IdP Web Server for Artifact Single Sign-on](#) (see page 138).

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

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

Follow these steps:

1. Create a server-side certificate request.
2. Have the Certificate Authority sign the server-side certificate.
3. Specify the server-side certificate in the web server configuration.
For the IIS Web Server used in the sample network, the IIS Certificate Wizard would be used.
4. [Enable a Persistent Session to Store Assertions at the IdP](#) (see page 138).

Enable a Persistent Session to Store Assertions at the IdP

Enable a persistent session for the realm that contains the authentication URL that you protected according to the instructions in [Protect the Authentication URL](#) (see page 113). The persistent session is required to store assertions for SAML artifact binding.

If you did not already enable a persistent session when you set up the authentication URL protection, follow this procedure for SAML artifact binding.

To enable a persistent session

1. Log in to the FSS Administrative UI.
2. From the Domains tab, expand the domain that contains the realm with the authentication URL, then expand the Realms object.

3. From the Realms List, select the realm with the authentication URL and from the menu bar select Edit, Properties of Realm.

The Realm Properties dialog opens.

4. Select the Session tab.
5. Click the Persistent Session.
6. Click OK.
7. [Select the Artifact Binding at the IdP](#) (see page 140).

Permit Access to the FWS Policy that Protects the Artifact Resolution Service

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

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

Follow these steps: at the IdP

1. Log on to the FSS Administrative UI.
2. Select the System tab.
3. From the menu, select Edit, Create Agent.
4. In the Agent Properties dialog, enter a name for the Web Agent then click OK. In this deployment, the Web Agent is idp-webagent.
5. If you do not have Agent Groups displayed, select View, Agent Groups from the menu bar.
6. Double-click the FederationWebServicesAgentGroup entry to open the Properties of Agent Group dialog.
7. Click Add/Remove and the Available Agents and Groups dialog opens.
8. Add idp-webagent, the IdP Web Agent protecting the FWS application, to the Agent group, by selecting it from the Available Members list and clicking the left arrow to move it to the Current Members list.
9. Click OK until you exit the Agent Groups dialog.
10. Specify that all the Service Providers under the affiliate domain Federation Sample Partners can access the artifact resolution service, as follows:
 - a. Select the Domains tab and expand FederationWebServicesDomain.
 - b. Select Policies.

- c. From the Policy List, double-click the SAML2FWSArtifactResolutionServicePolicy entry.
The SiteMinder Policy dialog opens.
- d. From the Users tab, select the SAML2FederationCustomUserStore tab then click Add/Remove.
affiliate: Federation Sample Partners is the "user" listed in the Available Members list.
- e. From the Available Members list, select the SP Partners domain and move it to the Current Members list, then click Apply.
- f. Click OK to return to the Policy List.

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

Select the Artifact Binding at the IdP

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

To configure artifact single sign-on

1. Log in to the FSS Administrative UI.
2. From the Domains tab, expand Federation Sample Partners and select SAML Service Providers to display the Service Providers.
3. Select sp.demo and right-click to open the properties of this dialog.
4. Select the SSO tab.
5. Complete the following fields:

Audience

sp.demo

This value must match the value at the Service Provider.

Assertion Consumer Service

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

6. Select the HTTP-Artifact check box.
7. For the Artifact encoding, select URL.
The artifact is added to a URL-encoded query string.

8. Complete the password fields:

Password

smfederation

Confirm Password

smfederation

The sp.demo uses this password to access the Federation Web Services application at the Identity Provider. This value must also match the value at the Service Provider.

9. For the Authentication Level, Validity Duration, and AuthnContext Class Ref fields, accept the defaults.

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

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

10. Click OK.
11. [Add a CA Certificate to the Smkeydatabase at the SP](#) (see page 141).

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

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

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

A set of common root certificates are shipped with the default smkeydatabase. To use root certificate for web servers that are *not* in the key store, import the necessary root certificates into the smkeydatabase.

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

Use the SiteMinder smkeytool utility to modify the database.

To add a certificate to the smkeydatabase

1. Open a command window.
2. Verify that the Certificate Authority certificate is already in the database by entering:

```
smkeytool -listcerts
```

Look for an entry type of CertificateAuthorityEntry.
3. If the CA certificate is not present, import a new CA certificate by entering:

```
smkeytool -addCert -alias <alias> -infile <cert_file> -trustcacert
```

For this deployment, the command is:

```
smkeytool -addCert -alias sampleAppCertCA -infile docCA.crt -trustcacert
```
4. When asked if you trust the certificate, enter YES.
The certificate is added to smkeydatabase.
5. Restart the Policy Server to see the smkeydatabase changes immediately.
6. [Enable the Artifact Binding for SAML Authentication at the SP](#) (see page 142).

Enable the Artifact Binding for SAML Authentication at the SP

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

To specify artifact binding for the authentication scheme

1. Log on to the FSS Administrative UI.
2. From the System tab, select Authentication Schemes.
3. Select Partner IdP.demo Auth Scheme and right-click to open the properties for this scheme.
4. Click Additional Configuration.
5. Select the SSO tab.
6. On the SSO tab, check HTTP-Artifact and enter the following value for the Resolution Service field:

```
https://www.idp.demo:443/affwebservices/saml2artifactresolution
```
7. Select the Backchannel tab and complete the following fields:

Authentication

Basic

SP Name

sp.demo

Password

password

Confirm Password

password

The password must match at the Identity Provider.

8. Click OK.
9. [Add a Link at the SP to Initiate Artifact Single Sign-on](#) (see page 143)

Test Artifact Single Sign-on

Test single sign-on in a SiteMinder-to-SiteMinder network using the web pages included with the sample application. The sample web pages are available provided you run the sample application script. If you do not run the sample application, use your own web pages to test single sign-on.

The sample application web pages are located in the following two folders.

policy_server_home/samples/federation/content/idpsample

policy_server_home/samples/federation/content/spsample

policy_server_home

Specifies the installed location of the SiteMinder Policy Server

Important! If you have run the sample application, the *idpsample* and *spsample* folders are automatically copied into the document root directory of your web server.

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

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

server:port

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

IdP_ID

Defines the provider ID.

The link for this deployment is:

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

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

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

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

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

Now, follow the steps to [test SP-initiated single sign-on](#) (see page 131).

Include an Attribute in the Assertion

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

For this deployment, an attribute is added for user1.

To add the firstname attribute

1. Log in to the FSS Administrative UI.
2. Select the Attributes tab from the SAML Service Provider Properties dialog.
3. Click Create.

The SAML Service Provider Attribute dialog opens.

4. Complete the following fields:

Attribute

unspecified (default)

Attribute Kind

User Attribute

Variable Name

firstname

Attribute Name

givenname

givenname is a attribute in the profile of user1.

5. Click OK to save your changes and return to the Attributes tab.

Configure Digital Signing (required for POST Binding)

For POST single sign-on, the SAML response must be signed. The configuration tasks at the Identity Provider and Service Provider enable digital signing.

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

Required task at the Identity Provider:

- [Add a Private Key and Certificate to the IdP SMkeydatabase](#) (see page 145)

Required tasks at the Service Provider:

- [Set Up the Key Database at the SP to Validate Digital Signatures](#) (see page 146)
- [Enable Signature Validation at the SP](#) (see page 147)

Add a Private Key and Certificate to the IdP Smkeydatabase

Keys and certificates used to sign SAML assertions for POST binding are stored in the smkeydatabase. Signing a SAML response is required, so create smkeydatabase at the Identity Provider and add the appropriate items to it.

If you deployed the sample application, you can use the key that it automatically installs. If you want to create a new key, use the smkeytool utility to delete all the data from the smkeydatabase and complete the following procedures.

To create a key database and add a private key and certificate to it

1. Open a command window.
2. If necessary, create a key database for a Windows system by entering
smkeytool.bat –createDB -password password

This command creates the smkeydatabase.

3. Add a private key and certificate to smkeydatabase.

idp.demo signs the SAML response before sending it to sp.demo.

The command for this deployment is:

```
smkeytool.bat -addPrivKey -alias defaultenterpriseprivatekey -keyfile  
"c:\program  
files\ca\siteminder\certs\post-pkey.der" -certfile "c:\program  
files\ca\siteminder\certs\post-cert.crt" -password password
```

The first part of this command is the location of the private key in DER format at the Identity Provider. For this deployment, that is post-pkey.der. The second part of the command is the location of the public key certificate, which is post-cert.crt followed by the password associated with the private key, which is password.

4. Restart the Policy Server to see the smkeydatabase changes immediately.
5. Log in to the FSS Administrative UI.
6. From the Domains tab, select Federation Sample Partners, then open the properties for the Service Provider, sp.demo.
7. Go to the General tab in the SAML Service Provider Properties dialog.
8. Uncheck the box labeled Disable Signature Processing. Deselecting this check box means that signature processing is enabled.
9. Click OK.
10. [Set Up the smkeydatabase at the SP to Validate Digital Signatures](#) (see page 146).

Set Up smkeydatabase at the SP for Signature Validation

For POST single sign-on, the Identity Provider digitally signs the SAML assertion, as required by the SAML 2.0 specification. Consequently, the Service Provider must validate the signature.

To validate a digital signature, add a public key to the smkeydatabase file of the Service Provider. When you configure the SAML authentication scheme, you specify the DN of the issuer and serial number of the corresponding partner certificate.

To import the public key

1. Open a command window.
2. Create the smkeydatabase by entering:

```
smkeytool.bat -createDB -password password
```

This command creates the smkeydatabase at the Service Provider with the password federation.

3. Add the public key certificate to smkeydatabase by entering:

```
smkeytool.bat -addCert -alias <alias> -infile path_to_X.509_certificate_file
```

In this deployment, the public key is post-cert.crt. The command is:

```
smkeytool.bat -addCert -alias idp1cert -infile "c:\program files\  
ca\siteminder\certs\post-cert.crt"
```

4. Restart the Policy Server to see the smkeydatabase changes immediately.
5. [Enable Signature Validation at the SP](#) (see page 147).

Enable Signature Validation at the SP

To validate a digital signature for POST single sign-on

1. Log in to the FSS Administrative UI.
2. From the System tab, select Authentication Schemes to display the Authentication Scheme List.

Select the existing SAML 2.0 authentication scheme, Partner IdP.demo Auth Scheme

The Authentication Scheme Properties dialog opens.

3. In the Scheme Common Setup section, clear the Disable Signature Processing. Disabling this option enables signature processing.
4. In the D-Sig Info box, enter the following:

Issuer DN

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

Serial Number

008D 8B6A D18C 46D8 5B

The D-Sig information enables the Service Provider to verify the SAML response signature. The values for the Issuer DN and Serial Number are from the public key in the smkeydatabase of the Service Provider.

5. Click OK.
Validation configuration is now complete.
6. Test POST single sign-on.

Encrypt and Decrypt the Assertion

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

The Identity Provider encrypts the assertion with the public key, which corresponds to the private key and certificate that the Service Provider uses to decrypt the assertion.

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

Required tasks at the IdP:

- [Add a Public Key to SMkeydatabase at the IdP](#) (see page 148)
- [Enable Encryption in the Policy Server User Interface at the IdP](#) (see page 148)

Required task at the SP:

- [Decrypt an Encrypted Assertion at the SP](#) (see page 149)

Add a Public Key to Smkeydatabase at the IdP

In this deployment, sp_encrypt.crt is the public key.

To add the public key to the IdP smkeydatabase

1. Open a command window.
2. Add the public key to the smkeydatabase by entering:

```
smkeytool -addCert -alias idp1 -infile "c:\program files\ca\siteminder\certs\sp-encrypt.crt"
```
3. Restart the Policy Server to see the smkeydatabase changes immediately.

Enable Encryption in the Policy Server User Interface at the IdP

To enable encryption at the IdP

1. Log on to the FSS Administrative UI.
2. From the Service Provider Properties dialog, select the Encryption tab.
3. Select the Encrypt Assertion.
4. Accept the defaults for the Encryption Block Algorithm and the Encryption Key Algorithm.

5. In the Issuer DN, enter the issuer of the Service Provider public key. In this deployment, the public key is sp-encrypt.crt.

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

Note: The value you enter for the Issuer DN field should match the issuer DN of the certificate in the smkeydatabase. We recommend you to open a command window and enter the command `smkeytool -listCerts` to list the certificates. View the DN to verify at you enter a matching value.

6. In the Serial Number field, enter the serial number of the public key that resides in the smkeydatabase of the Identity Provider. In this deployment, the value is 00EFF6AFB49925C3F4

The number must be hexadecimal.

7. Click OK to save your changes.
8. [Decrypt an Encrypted Assertion at the SP](#) (see page 149).

Decrypt an Encrypted Assertion at the SP

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

The Service Provider accepts an encrypted assertion from the Identity Provider as long as it has the private key and certificate to decrypt the assertion. You do not have to enable the Require an Encrypted Assertion feature for the SAML authentication scheme to accept an encrypted assertion at the Service Provider.

To add the private key and certificate to the smkeydatabase

1. Open a command window.
2. Do *one* of the following:
 - If the smkeydatabase has not been created already, enter the command:
`smkeytool.bat -createDB -password fedDB`
This command creates the smkeydatabase at the Service Provider with the password fedDB.
 - If smkeydatabase does exist, skip to the next step.

3. Add a private key and certificate to the existing smkeydatabase.

The command for this deployment is:

```
smkeytool.bat -addPrivKey -alias sp1privkey -keyfile "c:\program
files\ca\siteminder\certs\sp-encrypt.der" -certfile "c:\program
files\ca\siteminder\certs\sp-encrypt.crt" -password fedsvcs
```

The first part of this command is the location of the private key, sp-encrypt.der. The second part of the command is the location of the public key, sp-encrypt.crt, followed by the password, fedsvcs. Fedsvcs is the password associated with the private key.

4. Restart the Policy Server to see the smkeydatabase changes immediately.
5. Test single sign-on. Go to either of the following:
 - Test SAML 2.0 POST Single Sign-on
 - Test Artifact Single Sign-on

Chapter 4: Overview of a SiteMinder Federation Setup

This section contains the following topics:

[Installation Overview](#) (see page 151)

[Conventions in the Installation Overview Procedures](#) (see page 152)

[Set Up Asserting Party Components](#) (see page 153)

[Set Up Relying Party Components](#) (see page 161)

Installation Overview

This overview outlines the set up of a SiteMinder federated network.

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

These procedures refer to the latest <stmndr releases. For other compatible versions, see the SiteMinder Platform Matrix associated with the release.

To locate the SiteMinder Platform Support Matrix

1. Log on to the [Technical Support Site](#).
2. Search for SiteMinder Platform Support Matrix.

Be aware of the following:

- SiteMinder does not support federation between two systems using the same cookie domain.
- This overview does not include the SAML Affiliate Agent. For information involving that Agent, see the *SiteMinder SAML Affiliate Agent Guide*.
- Federation Security Services and the SAML Affiliate Agent are separately licensed items from SiteMinder.

Conventions in the Installation Overview Procedures

The following variables are used in installation and configuration procedures:

web_agent_home

Specifies the installed location of the Web Agent

policy_server_home

Specifies the installed location of the Policy Server

web_server_home

Indicates the installed location of the web server

fqn

Designates fully qualified host name

port_number

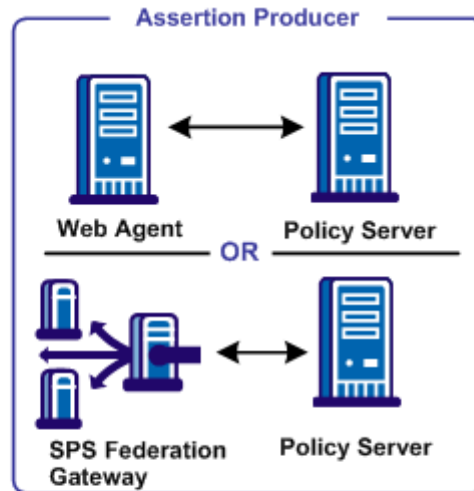
Specifies the port number of a server

sps_home

Specifies the installed location of CA SiteMinder SPS

Set Up Asserting Party Components

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



1. Install the Policy Server
Policy Server Installation Guide
2. Set up affiliate domains and affiliates/SPs/RPs
3. Install and configure a Web Agent or SPS Federation Gateway (skip steps 4 and 5 if using SPS)
Web Agent Installation Guide or SPS Administration Guide
4. Install a web or application server for the Web Agent Option Pack
5. Install the Web Agent Option Pack
Web Agent Option Pack Guide
6. Configure Federation Web Services
7. Protect Federation Web Services
8. For SAML 2.0 responses, signing is required
9. Create links to target resources at the consumer/SP

Except where noted, see this guide for configuration instructions

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

Install the Policy Server at the Asserting Party

The setup at the asserting party is as follows:

1. Install the Policy Server.

Policy Server Installation Guide.

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

Policy Server Administration Guide.

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

3. Set up a policy store for use by the Policy Server.

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

Policy Server Installation Guide.

4. Set up a user directory.

Policy Server Configuration Guide.

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

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

Set up Affiliate Domains and Add Sites to these Domains

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

Follow these steps:

1. Access the FSS Administrative UI.
2. Create an affiliate domain.
3. Add a user store for users that the asserting party generates assertions.
4. Add an object for each relying party to the affiliate domain.

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

5. After you add sites to an affiliate domain, verify that you protect the AuthenticationURL. This verification affirms that a user has a session at the asserting party prior to process a request for a federated resource.

To do this task:

- a. Create a policy domain.
- b. Protect the policy domain with the Web Agent that is protecting the server with the Web Agent Option Pack.
- c. To this policy domain, add a realm, rule, and policy that protects the Authentication URL.

More Information:

[Add Entities to an Affiliate Domain](#) (see page 222)

[Authenticate Users with No SiteMinder Session \(SAML 1.x\)](#) (see page 227)

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

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

At the asserting party, set up the following components:

1. Install one of the following components:
 - Web Agent
For instructions, see the *Web Agent Installation Guide*.
 - SPS federation gateway
For instructions, see the *Secure Proxy Server Administration Guide*.
2. For artifact single sign-on, SSL-enable the web server with the Web Agent installed or the system with the SPS federation gateway.

If the SAML Affiliate Agent is the consumer, configure the SSL-enabled web server at the producer to ignore client certificates. The Web Agent is installed on this web server. If the web server is configured to accept client certificates, the affiliate server component of the SAML Affiliate Agent cannot communicate with the Web Agent.

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

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

At the asserting party:

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

Install the Asserting Party Web Agent Option Pack

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

At the asserting party:

1. Install the Web Agent Option Pack.

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

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

For the supported JDK version, log on to the [Technical Support site](#) and search for the SiteMinder Platform Support Matrix for the release.

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

Configure Federation Web Services (Asserting Party)

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

To configure Federation Web Services at the asserting party

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

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

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

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

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

web_agent_home

Represents the installed location of the Web Agent

sps_home

Represents the installed location of the SPS federation gateway

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

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

fqhn

Defines the fully qualified host name.

port_number

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

For example:

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

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

Assertion Retrieval Service has been successfully initialized.

The requested servlet accepts only HTTP POST requests.

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

Allow Access to Federation Web Services (asserting party)

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

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

[Grant access to specific FWS policies](#) (see page 179) that apply to features configured for your federation partnership.

Enable the Signing of SAML POST Responses

To sign SAML POST responses, which is required by the SAML specification, add a private key and certificate to the SiteMinder key database file, `smkeydatabase`.

Create Links to Target Resources (optional)

Go to one of the following:

- [Links for SAML 1.x Single Sign-On](#) (see page 159)
- [Links for SAML 2.0 Single Sign-On at the Identity Provider](#) (see page 160)
- [Links to Initiate WS-Federation Single Sign-on](#) (see page 160)

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

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

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

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

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

producer_site

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

consumer_site

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

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

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

producer_site

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

consumer_site

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

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

More Information:

[Creating Links to Consumer Resources for Single Sign-on](#) (see page 251)

Initiate SAML 2.0 Single Sign-On at the Identity Provider

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

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

```
http://idp_server:port/affwebservices/public/saml2sso?SPID=SP_ID
```

idp_server:port

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

SP_ID

Service Provider ID value.

Add the [ProtocolBinding query parameter](#) (see page 325) to this link depending on which bindings are enabled.

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

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

More information:

[Set Up Links at the IdP or SP to Initiate Single Sign-on](#) (see page 323)

[Unsolicited Response Query Parameters Used by a SiteMinder IdP](#) (see page 325)

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

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

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

The syntax for the link is:

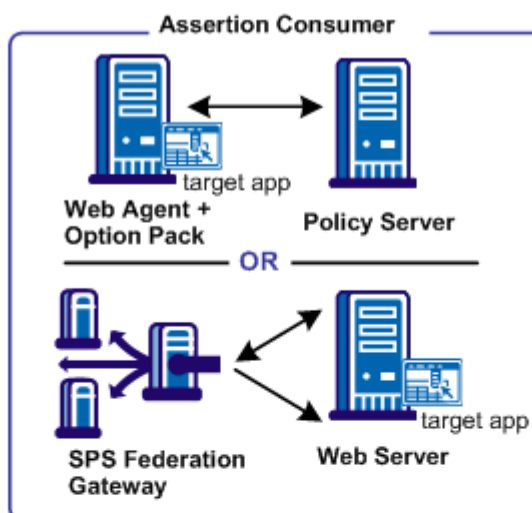
`https://AP:port/affwebservices/public/wsfedsso?wa=wsignin1.0&wtrealm=RP_ID`

ap_server:port

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

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

Set Up Relying Party Components



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

Except where noted, see this guide for configuration instructions.

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

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

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

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

Install the Relying Party Policy Server

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

At the relying party, do the following:

1. Install the Policy Server.

See the *SiteMinder Policy Server Installation Guide*.

2. Set up a policy store.

See the *SiteMinder Policy Server Installation Guide*.

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

3. Set up a user store and add users permitted to access target resources.

See the *SiteMinder Policy Server Configuration Guide*.

Configure a SAML or WS-Federation Authentication Scheme

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

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

Specifically:

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

More Information:

[Configure SiteMinder as a SAML 1.x Consumer](#) (see page 255)

[Configure SiteMinder as a SAML 2.0 Service Provider](#) (see page 347)

[Configure SiteMinder as a Resource Partner](#) (see page 425)

Protect Target Resources at the Relying Party

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

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

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

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

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

At the relying party, set up the following components:

1. Install one of the following components:
 - Web Agent
For instructions, see the *Web Agent Installation Guide*.
 - SPS federation gateway
For instructions, see the *Secure Proxy Server Administration Guide*.
2. Configure the Web Agent or SPS federation gateway.

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

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

At the relying party:

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

Install the Web Agent Option Pack at the Relying Party

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

At the relying party:

1. Install the Web Agent Option Pack.

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

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

To determine the required JDK version, go to the [Technical Support site](#) and search for SiteMinder Platform Matrix.

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

Configure Federation Web Services at the Relying Party

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

To configure Federation Web Services at the relying party

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

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

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

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

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

web_agent_home

Represents the installed location of the Web Agent

sps_home

Represents the installed location of the SPS federation gateway

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

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

fqhn

Defines the fully qualified host name.

port_number

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

For example:

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

If Federation Web Services is operating correctly, the following message appears:
Assertion Retrieval Service has been successfully initialized.
The requested servlet accepts only HTTP POST requests.

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

More Information:

[Configure Federation Web Services \(Asserting Party\)](#) (see page 157)

Allow Access to Federation Web Services (Relying Party)

The procedure for protecting the Federation Web Services application is the same at the relying party as it is for the asserting party.

More Information:

[Allow Access to Federation Web Services \(asserting party\)](#) (see page 158)

Set-up the smkeydatabase for Artifact Single Sign-on (optional)

The smkeydatabase is a local flat-file key database that stores keys and certificates needed for PKI specific operations such as encryption, decryption, signing, verification and client authentication.

For artifact single sign-on, the key database at the asserting party holds the certificate of the certificate authority used to establish an SSL connection. The SSL connection secures the back channel that the assertion is sent across for artifact single sign-on.

A set of common root CAs are shipped in the default smkeydatabase. To use root CAs for web servers that are *not* in smkeydatabase, import these root CAs into the file.

To modify smkeydatabase, use the smkeytool utility.

Create Links to Initiate Single Sign-on (optional)

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

More Information:

[Initiate SAML 2.0 Single Sign-on at the SP \(optional\)](#) (see page 167)

[Initiate WS-Federation Single Sign-on at the Resource Partner](#) (see page 168)

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

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

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

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

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

`http://SP_site/affwebservices/public/saml2authnrequest?ProviderID=IdP_ID`

sp_server:port

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

IdP_ID

Specifies the identity that is assigned to the Identity Provider.

You can add the ProtocolBinding query parameter to this link depending on which bindings are enabled. For more information about configuring links at the Service Provider, see [Set Up Links at the IdP or SP to Initiate Single Sign-on](#) (see page 323).

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

You can also create links at the Identity Provider.

More Information:

[Set Up Links at the IdP or SP to Initiate Single Sign-on](#) (see page 323)

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

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

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

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

The syntax for the link is:

`https://host:port/affwebservices/public/wsfedso?wa=wsignin1.0&wrealm=RP_ID`

host:port

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

RP_ID

Specifies the Resource Partner identity

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

Chapter 5: Setup the SAML 1.x Assertion Generator File

This section contains the following topics:

[SAML 1.x Assertion Generator Properties File](#) (see page 171)

SAML 1.x Assertion Generator Properties File

The Policy Server installed at the producer, includes a component named the assertion generator. The assertion generator creates SAML assertions.

The `AMAssertionGenerator.properties` file (SAML 1.x only) is required for the assertion generator to generate SAML 1.x assertions. After a SAML 1.x assertion is generated, the session store stores the assertion until the consumer requests it. This file also contains commented instructions, which you can read for more information about the settings in the file.

The installed location of this file is:

`policy_server_home/config/properties`

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

Configure the SAML 1.x `AMAssertionGenerator.properties` File

To configure the `AMAssertionGenerator.properties` file

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

AssertionIssuerID

Specifies the URL that identifies the site issuing the assertion.

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

Note: Set this value properly so that SAML 1.x assertions are meaningful.

SecurityDomain

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

SourceID

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

The values you enter in this file must match the values for the equivalent settings at the consumer site. The settings must match whether the consumer is a SAML Affiliate Agent or a 1.x consumer.

Note: If you update the AmAssertionGenerator.properties file, the Policy Server does not pick the changes until it is restarted.

Chapter 6: Review the JVMOptions File Which Creates a JVM

The JVMOptions.txt File

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

During a Policy Server upgrade, the existing JVMOptions.txt file is renamed to JVMOptions.txt.backup. A new JVMOptions.txt file is created.

If the original file included customized parameters, be sure to modify the newly created file to include these customized parameters.

The installed location of this file is:

policy_server_home/config/

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

Notes:

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

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

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

Chapter 7: Storing User Session, Assertion, and Expiry Data

This section contains the following topics:

[Federation Data Stored in the Session Store](#) (see page 175)

[Enable the Session Store](#) (see page 176)

[Environments that Require a Shared Session Store](#) (see page 177)

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

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

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

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

- Assertion Attributes Persistence (all profiles)

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

- Single logout (SAML 2.0)

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

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

- Sign-out (WS-Federation)

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

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

Enable the Session Store

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

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.

To configure a database for the session server

1. Select Session Server from the Database drop-down list.
2. Select an available storage type from the Storage drop-down list.
3. Set the Enable Session Server option.

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.

4. Specify Storage Options appropriate for the chosen storage type.
5. Click OK to save the settings and exit the Console.
6. Stop and restart the Policy Server.

Environments that Require a Shared Session Store

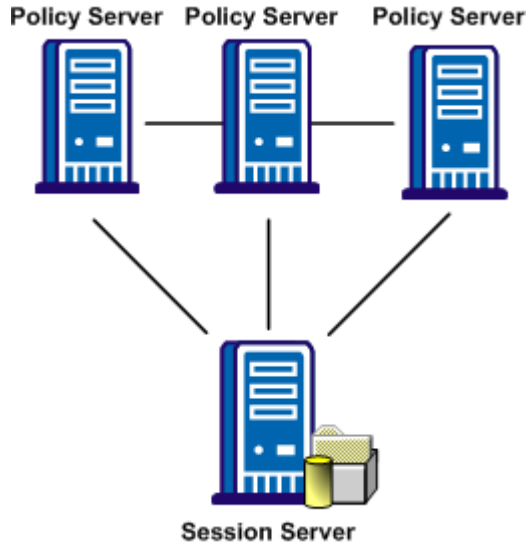
The following SiteMinder features require a shared session store to store SAML assertions and user session information.

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

- Configure the login realm for persistent sessions for all features *except* for an HTTP-POST single use policy.
Persistent sessions are part of the realm configuration.
- For HTTP-Artifact single sign-on, share the session store at the Producer/Identity Provider site across all Policy Servers in the cluster.
Sharing the session store verifies that all Policy Servers have access to assertions when each one receives a request for an assertion.
- For SAML 2.0 single logout and WS-Federation signout, share the session store at the asserting and relying party across all Policy Servers in the cluster.
Sharing the session store verifies that all Policy Servers have access to user session data when each one receives a request for a session logout.
- For the HTTP-POST and WS-Federation single use policy feature, share the session store at the relying party across all Policy Servers in the cluster.

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

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



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

- Point all Policy Servers to one session store
In the Policy Server Management Console, configure the Policy Server to use the designated session store.
- Replicate the session store across many session stores.
For instructions on replicating a database, use the documentation for your database.

Chapter 8: Grant Access to Federation Web Services

This section contains the following topics:

[Policies that Protect Federation Web Services](#) (see page 179)

[Features Associated with FWS Policies](#) (see page 180)

[Enforce the Policies that Protect Federation Web Services](#) (see page 181)

Policies that Protect Federation Web Services

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

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

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

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

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

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

Features Associated with FWS Policies

The policies that SiteMinder creates support the following Federation Security Services features:

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

Enforce the Policies that Protect Federation Web Services

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

Granting access involves the following tasks:

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

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

Detailed procedures for enforcing the HTTP-Artifact assertion retrieval and attribute authority policies are in the relevant sections for those features. Procedures for allowing access to the notification and session service policies are similar. These services are relevant only if a SAML Affiliate Agent is a consumer.

More information:

[Grant Access to the Service for Assertion Retrieval \(Artifact SSO\)](#) (see page 234)

Chapter 9: Signing and Encrypting Messages to Secure Federated Transactions

This section contains the following topics:

[Certificate and Private Key Usage for Federation](#) (see page 183)

[SmKeyDatabase Overview](#) (see page 186)

[Formats Supported by the Smkeydatabase](#) (see page 192)

[Properties File for the Key Database](#) (see page 192)

[Modify the Key Database Using smkeytool](#) (see page 195)

[Migrate AM.keystore and Update smkeydatabase](#) (see page 208)

Certificate and Private Key Usage for Federation

SiteMinder uses private key/certificate pairs for signing, verification, encryption, and decryption of entire assertions, or specific assertion content. Federation Security Services also employs client certificates for authenticating a client across a back channel for artifact single sign-on. Finally, it uses SSL server certificates for establishing SSL connections.

There can be multiple private keys/certificate pairs in the SiteMinder key database, named the smkeydatabase. If you have multiple federated partners, you can use a different pair for each partner.

Private key/certificate pairs and single certificates are stored in the SiteMinder key database. Each key/certificate pair, client certificate, and trusted certificate in the key database must have a unique alias. The alias enables SiteMinder to reference any key or certificate in the key database. You can manage the contents of the key database using the smkeytool utility.

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

Function	Private Key/Cert Pair	Certificate (public key)	SSL Server Certificate	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				
Secures SSL connections			X		
Serves as credential for client certificate authentication of the artifact back channel					X
Validates other certificates and CRLs				X	

The following sections detail key and certificate use for federated communication.

Signing and Verification Operations

SiteMinder 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/Decryption Operation

For SAML 2.0 you can configure federation security services to encrypt an entire assertion, the NameID, or other attributes. If you enable encryption, the asserting party uses the certificate (public key) sent by the relying party 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 does not support encryption of assertion data.

Certificates for SSL Connections

You can enable SSL for the artifact back channel or for general federated communication. Establishing the SSL connection requires that the relying party has the CA certificate associated with the signed SSL server certificate. The SSL server certificate secures the SSL connection, and the CA certificate verifies the SSL server certificate is trusted.

Client Certificate Authentication Across the Back Channel

For artifact single sign-on, assertions are sent across a back channel. One way to secure the back channel is to require that the relying party provide a client certificate as its credential. This credential lets the relying party gain access to the Artifact Resolution Service at the asserting party.

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 SiteMinder 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. SiteMinder sends the response back to the relying party over a back channel, which is a secured connection between the asserting and relying party. In contrast, web browser communication occurs over the front channel.

You can 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

For any of these authentication methods, the relying party back channel must be configured so it can communicate with the Assertion Retrieval Service (SAML 1.1) or the Artifact Resolution Service (SAML 2.0) in a secure manner.

The following considerations might be useful when choosing an authentication method for the artifact back channel:

- We recommend that you use an SSL connection for the back channel. The SSL connection should be secured by an SSL server certificate signed by a trusted CA.

A set of common root and intermediate CA certificates are shipped with the default key database. To use a server certificate signed by a CA that is not already in the key store, you must import the CA certificate into the database as a trusted CA certificate.

Federation uses an SSL-client when processing back channel requests. You can configure the IdP Web server to use SSL versions TLSV1_1 and TLSV1_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 IdP server 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-based authentication will fail. This certificate must also be imported to smkeydatabase at the asserting party, since it is compared as part of the authentication. The client certificate serves as the relying party credentials to access the Assertion Retrieval Service or Artifact Resolution Service when the service is protected by client certificate authentication.

SmKeyDatabase Overview

The smkeydatabase is a key and certificate database used for signing, verification, encryption, and decryption between a SiteMinder asserting party and a SiteMinder relying party. The database is made up of multiple files. You can manage and retrieve keys and certificates in this database using the SiteMinder tool called smkeytool.

The smkeydatabase can store multiple private keys and certificates. If you have multiple federated partners, you can use a different private key for each partner. Every certificate or key/certificate pair must have a unique alias. The alias enables you to reference any single certificate or key/certificate pair in the smkeydatabase.

If a signing alias is configured, SiteMinder uses the key associated with that alias to sign assertions. If no signing alias is configured, SiteMinder uses the key with the alias **defaultenterpriseprivatekey** to sign assertions. If there is no default enterprise private key found, then SiteMinder uses the first private key that it finds in the database to sign assertions.

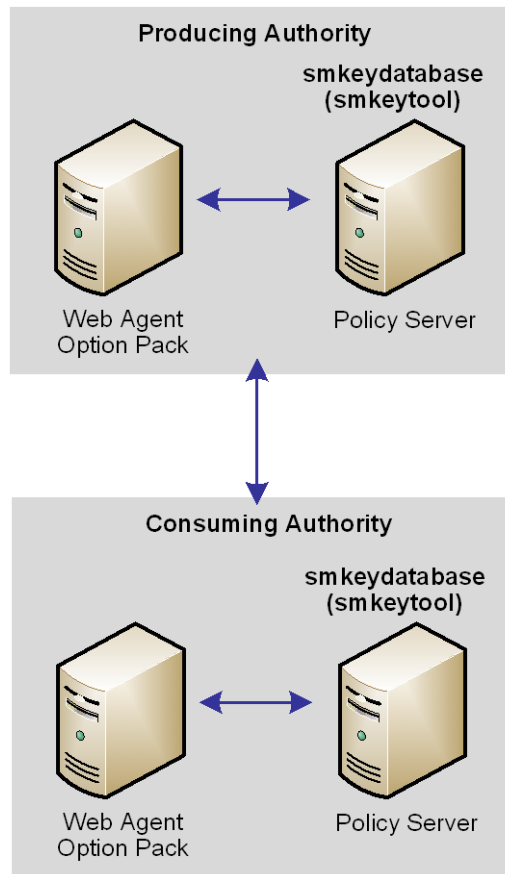
Important! To store multiple keys in the database, you must define the first key you add with the alias defaultenterpriseprivatekey before you can add subsequent keys.

A given Policy Server may sign and/or verify responses. Keys and certificates for signing and validation can be added to the same key database, depending on what the Policy Server is doing. For single sign-on, if a site is only consuming assertions using SAML POST profile, then that consumer/Service Provider only verifies the response; it never signs it. In the case of single logout, it depends upon which site initiates the single logout that determines which side signs or verifies requests and responses.

The smkeydatabase is installed with a SiteMinder Policy Server. The Policy Server uses certified Federal Information Processing Standard (FIPS) 140-2 compliant cryptographic libraries, which enables a SiteMinder environment to use FIPS-compliant algorithms to encrypt sensitive data. As a result, all data in the smkeydatabase is encrypted using these FIPS-compliant algorithms.

Note: If you upgrade from a previous version of the Policy Server to r12.0 SP3, see the *SiteMinder Upgrade Guide* for instructions on migrating the smkeydatabase so that data is properly encrypted.

The following illustration shows the location of the key store in a SiteMinder federated network.



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

Role of the Smkeydatabase at the Asserting Party

At the asserting party, the smkeydatabase is used for the following features:

- Single sign-on (SAML 1.x POST profile, SAML 2.0 POST binding, or WS-Federation)

For SAML 1.x POST binding, SAML 2.0 POST binding or WS-Federation Passive Requester Profile, the asserting party needs to sign the SAML the assertion. The relying party that receives the assertion verifies that signature.

- Encryption of assertions, Name IDs and attributes for SAML 2.0 artifact or POST authentication

If you enable encryption, the asserting party must provide the public key certificate of the Service Provider for encrypting the data, while the relying party uses a private key to decrypt the data.

- Single logout

For single logout, the side initiating the logout request signs the request and the side receiving the request validates the signature. Conversely, the receiving side must sign the response and the initiator must validate the response.

- AuthnRequests for Single sign-on

The Identity Provider can require that the Service Provider sign AuthnRequest messages. To sign these messages, you have to have a private key and certificate. The Identity Provider then needs to validate the request with the public key that corresponds to the private key.

To accomplish signing, verification, and encryption, you must set up an smkeydatabase for each Policy Server that is responsible for signing, verification, and encryption.

Role of the Smkeydatabase at the Relying Party

At the relying party, the smkeydatabase is used for SAML 1.x and SAML 2.0 artifact single sign-on.

For SAML 1.x and SAML 2.0 artifact protocol, the relying party sends a request for the assertion to the Assertion Retrieval Service (SAML 1.x) or the Artifact Resolution Service (SAML 2.0). These services retrieve the assertion from the asserting party, which then returns the assertion to the relying party over a back channel.

It is recommended that you protect these services from unauthorized access. To secure the Assertion Retrieval or Artifact Resolution Service, use one of the following authentication methods:

- Basic (SAML 2.0 Artifact Resolution Service only)
- Basic over SSL
- X.509 Client Certificate

For any of these authentication methods, the smkeydatabase at the relying party must be configured correctly so it can communicate with the Assertion Retrieval Service or Artifact Resolution Service in a secure manner.

If the connection between the two entities is an SSL connection, the relying party needs to have the Certificate Authority (CA) certificate associated with the server certificate from the asserting party to ensure that it trusts the server certificate. If an X.509 client certificate is required to establish a connection, then the smkeydatabase at the relying party must contain the client certificate.

Certificates Stored in the smkeydatabase Only at the Asserting Party

The following types of certificates are stored in smkeydatabase at the relying party site:

Certificate Authority (CA) certificates

Used for establishing an SSL connection from a relying party to the web server at a asserting party.

A set of common root CA certificates are shipped with the default smkeydatabase. To use a certificate for a CA that are not already in the key store, you must import the certificate into the database.

Client certificates

Used for sending a certificate from a relying party to an asserting party. The certificate serves as credentials when the consumer must authenticate using a client certificate authentication scheme to access the Assertion Retrieval or Artifact Resolution Service.

Partner certificates

Used for performing digital signature verification at the relying party to ensure the partner issuing the assertion is a trusted site. At a SAML 2.0 Identity Provider, the partner certificate is used to verify the signed messages from the Service Provider during single logout. The Service Provider certificate must exist at Identity Provider's machine.

When the Web Agent initializes, it gets all the client and server certificates, but the keys remain at the Policy Server.

Certificate Revocation Lists in the smkeydatabase

A Certificate Revocation Lists (CRL) is issued by a Certificate Authority to its subscribers. The list contains serial numbers of subscribers whose digital certificates have been revoked. When a user attempts to access a server, the server allows or denies access based on the CRL entry.

If Federation Security Services tries using a revoked partner certificate, you see a message in the SAML assertion that SAML authentication has failed.

If you are using CRLs, the smkeydatabase must point to a current CRL for each root CA certificate to help the Policy Server enforce secure access. To add and maintain a CRL in the smkeydatabase, a series of command options are available with SiteMinder's smkeytool utility, which is used to modify the smkeydatabase.

If you are using CRLs, you need to specify the location of a CRL for the smkeydatabase. Updating a CRL differs depending on the CRL type. To update a certificate file, you have to point the smkeydatabase to the most updated file. To update LDAP CRLs, the location of the list must be specified and then the server administrator can configure the list to be updated automatically.

Note: The CRL feature for the smkeydatabase has no relationship to the SiteMinder client certificate authentication scheme. Federation CRL features must be configured on their own.

The CRL feature for the smkeydatabase includes the following:

- Addition of certificate files or LDAP CRLs

Note: The Policy Server explicitly requests LDAP CRLs in binary transfer encoding, using the certificateRevocationList;binary or authorityRevocationList;binary LDAP attributes. Therefore, 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.

- PEM and DER encodings for file CRLs
- Only DER encodings for LDAP CRLs
- SAML 1.x, SAML 2.0, and WS-Federation protocols

The CRL feature does not support the Online Certificate Status Protocol (OCSP).

You can add a CRL to the smkeydatabase using smkeytool.

To add a CRL to the smkeydatabase

1. Add the CRL file or LDAP CRL to smkeydatabase with the addRevocationList command option.

Example:

```
smkeytool -addRevocationInfo -issueralias verisignca -type filecrl  
-location c:\crls\verisign_root_ca.crl
```

2. Restart the Policy Server.

Formats Supported by the Smkeydatabase

The smkeydatabase supports the following formats:

- Private Keys: Private keys must be in PKCS1, PKCS5, PKCS8 or PKCS12 format and DER or PEM encoded. Only RSA keys are supported.
- Public Certificates: V1, V2 and V3 of the X.509 certificate format are supported. DER, Base64, and PEM encoding formats are supported.

Properties File for the Key Database

The key database properties file, smkeydatabase.properties, defines the configuration parameters required to access and manage the key database.

The smkeydatabase.properties file is installed in:

- *siteminder_home*\config\properties (Windows)
- *siteminder_home*/config/properties (UNIX)

Modify this file only to change the following options:

- NativeDBName--specifies name of the key database
- DBLocation--indicates the directory where the key database resides
- DBUpdateFrequencyMinutes--specifies the frequency at which the database is read from the file system.

The smkeydatabase.properties file contains the following settings:

- [DBLocation](#) (see page 193)
- [NativeDBName](#) (see page 193)
- [XMLDocumentOpsImplementation](#) (see page 193)
- [AffiliateXMLSignatureImplementation](#) (see page 194)
- [IXMLSignatureImplementation](#) (see page 194)
- [EncryptedPassword](#) (see page 194)
- [IXMLEncryptDecryptImplementation](#) (see page 194)
- [DBUpdateFrequencyMinutes](#) (see page 195)
- [LDAPAccessTimeout](#) (see page 195)

Descriptions of each setting follow.

DBLocation Setting

Specifies the path to the directory where the database resides.

Enter the location that smkeytool should use when you manually create the database.

Default: *policy_server_home/smkeydatabase*

NativeDBName Setting

Identifies the name of the database.

Specify the name you want smkeytool to use when you create the database.

Default: smkeydatabase

XMLDocumentOpsImplementation Setting

Specifies the Java class that implements the XML signing and validation.

Note: Do not change this value; it is static and preconfigured.

Default: com.ca.smkeydatabase.api.XMLDocumentOpsImpl

AffiliateIXMLSignatureImplementation Setting

Specifies the Java class that implements low-level cryptographic operations for signing and validation.

Note: Do not change this value; it is static and preconfigured.

Default: com.ca.smkeydatabase.api.XMLSignatureApacheImpl

IXMLSignatureImplementation Setting

Specifies the Java class for Transactionminder that implements low-level cryptographic operations for signing and validation.

Note: Do not change this value; it is static and preconfigured.

Default: com.ca.smkeydatabase.api.XMLSignatureApacheImpl

EncryptedPassword Setting

Indicates the smkeydatabase password.

(Encrypted using the policy store key at database creation.) Prior to creating a key database, this entry contains a dummy value.

Default: *encrypted_password_string*

IXMLEncryptDecryptImplementation Setting

Identifies the Java class that implements the encryption and decryption of assertions, Name IDs, and attributes.

Note: Do not change this value; it is static and preconfigured.

Default: com.ca.smkeydatabase.api.XMLEncryptDecryptApacheImpl

DBUpdateFrequencyMinutes Setting

Indicates the frequency at which the database is read from the file system. Specifically, it is the number of minutes after which the in-memory smkeydatabase expires and is reloaded.

Until this interval passes, certificates and keys added, removed, or changed in the database will not affect the Policy Server. If the value is 0, key database caching is disabled entirely. If the value is -1, the cache persists until the Policy Server is restarted.

Default: 60 minutes

LDAPAccessTimeout

Sets the maximum number of seconds that SiteMinder waits for an LDAP server operation to complete before the connection times out. For LDAP CRL checking, this setting determines the amount of time SiteMinder waits for a response from the LDAP directory to examine the CRL.

Enter a positive integer. For example, if you enter 30, the timeout is 30 seconds.

If SiteMinder connects to an LDAP server in a high latency network, increase the LDAPAccessTimeout connection timeout. If your network requires faster LDAP access, decrease the timeout value.

Default: 60

Modify the Key Database Using smkeytool

Smkeytool is a SiteMinder command-line utility that manages the key database (smkeydatabase). The smkeytool utility is installed with the Policy Server in the following locations:

- *siteminder_home*/bin (UNIX)
- *siteminder_home*\bin (Windows)

Use smkeytool to:

- Create and delete a key database
You can only have one key database per Policy Server. After the database is created, you can add keys and certificates.
- Add and delete private keys
- Add and delete a partner certificate
- List all certificates stored in the key database

- Import root certificates of CAs
- Add client certificate keys

If you are using a root or chain Certificate Authority (CA) at the relying party that is not listed in the smkeydatabase, add it to the smkeydatabase.

For example, a signed VeriSign CA server-side certificate is used to SSL-enable the producer-side web server installed with the Web Agent Option Pack. To use this certificate for Basic over SSL authentication, add the VeriSign certificate to the smkeydatabase at the consumer. The addition of the certificate helps ensure that the consumer is communicating with a producer with a server-side certificate. The presence of the certificate also helps ensure that a trusted CA verified the certificate.

- Export key data from smkeydatabase
- Add, list, validate, and delete a Certificate Revocation List

Notes About Modifying Certificates

- If you are adding a private key/certificate pair or single certificate, delete the certificate metadata from the certificate file before trying to import it into the smkeydatabase. Import only the data starting with the --BEGIN CERTIFICATE-- marker and ending with the --END CERTIFICATE-- marker. Be sure to include the markers.
- If you add a new certificate to the key database or update an existing certificate, restart the Policy Server to see the change immediately. If you do not restart the Policy Server, it takes some time before the Policy Server and the key database synchronize. The amount of time for the key database to update depends on the configured frequency of database updates. You can configure database updates by adjusting the DBUpdateFrequencyMinutes parameter in the smkeydatabase.properties file.

Smkeytool Command Syntax and Options

Smkeytool is a command-line utility that provides many options to manage the smkeydatabase.

Run the smkeytool utility from a command line, using the following syntax:

UNIX

```
smkeytool.sh -option [-argument(s)]
```

Windows

```
smkeytool.bat -option [-argument(s)]
```

If you enter smkeytool from a command line without any options, you will see a list of all command line options.

The smkeytool utility uses the following command options and arguments:

- [createDB](#) (see page 198)
- [addPrivateKey](#) (see page 198)
- [addCertOption](#) (see page 199)
- [addRevocationInfo](#) (see page 200)
- [changepassword](#) (see page 201)
- [deleteRevocationInfo](#) (see page 201)
- [deleteDB](#) (see page 201)
- [delete](#) (see page 202)
- [export](#) (see page 202)
- [findAlias](#) (see page 203)
- [importDefaultCACerts](#) (see page 203)
- [listCerts](#) (see page 203)
- [listRevocationInfo](#) (see page 203)
- [printCert](#) (see page 204)
- [renameAlias](#) (see page 204)
- [validateCert](#) (see page 204)
- [help](#) (see page 204)

A description of each command option follows.

createDB Option

Creates a new smkeydatabase to store keys and certificates. By default, the directory is named smkeydatabase. You can change the smkeydatabase location by modifying the smkeydatabase.properties file.

All private keys in the smkeydatabase are encrypted using FIPS-compliant algorithms.

Important! To store multiple keys in the database, you must define the first key you add with the alias defaultenterpriseprivatekey before you can add subsequent keys.

Arguments for -createDB are as follows:

-password <password>

Required. The password is used to store all data in an encrypted format in the key database. It can be a value from 6 to 32 characters. It is encrypted using the policy store key and added to the smkeydatabase.properties file.

-importDefaultCACerts

(Optional) Imports the default Certificate Authority (CA) certificates during the creation of the database. These certificates are imported from the cacerts.keystore file, which is installed with the Policy Server and contains all default CA certificates. This option is the same as executing the -importDefaultCACerts option.

addPrivKey Option

Adds a private key/certificate pair to the key database. Use this command to import only a private key/certificate pair into the key database. You can have multiple private key/certificate pairs in the database, but SiteMinder supports only RSA keys in the database.

Note: Only private key/certificate pairs are stored in the smkeydatabase in encrypted form.

The Policy Server at the asserting party uses a single private key/certificate pair to sign SAML assertions and the certificate to decrypt encrypted SAML assertions received from the relying party. Typically, the key is the first private key/certificate pair found in the smkeydatabase.

With the -addPrivKey command, you can specify the key data by combining the -keyfile and -certfile options or by using the -keycertfile option alone.

Arguments for -addPrivKey are as follows:

-alias <alias>

Required. Assigns an alias to a private key/certificate pair in the database. The alias must be a unique string and can contain only alphanumeric characters.

-certfile <cert_file>

Specifies the full path to the location of the certificate associated with the private key/certificate pair. Required for keys in PKCS1, PKCS5, and PKCS8 format.

-keyfile <private_key_file>

Specifies the full path to the location of the private key file. Required for keys in PKCS1, PKCS5, and PKCS8 format.

-keycertfile <key_cert_file>

Specifies the full path to the location of the PKCS12 file that contains the private key/certificate pair data. Required for keys in PKCS12 format.

-password <password>

Optional. Specifies the password that was used to encrypt the private key/certificate pair when the pair was originally created. When a key/certificate pair is added to the smkeydatabase, supply this password to decrypt the pair before it gets written to the smkeydatabase.

Note: This password is not stored in the smkeydatabase.

After the key/certificate pair is decrypted and placed in the smkeydatabase, SiteMinder encrypts the pair again using its own password. The password SiteMinder uses is the one you specified when establishing the smkeydatabase.

addCert Option

Adds only a certificate to the key database. Use this command option only to import a public certificate. The certificate can be a certificate associated with a private key/certificate pair; however, you are only adding the certificate to the key database. You can also use this command to import trusted CA certificates.

Note: If you trust a certificate as a Certificate Authority, this certificate is always treated as a CA certificate.

For X.509 certificate formats, SiteMinder supports V1, V2, and V3 versions. For encoding formats, SiteMinder supports DER and PEM formats. Restart the Web Agent when you add a Certificate Authority certificate.

Arguments for addCert are as follows:

-alias <alias>

Required. Alias to the certificate associated with this private key in the database. Must be a unique string and should contain only alphanumeric characters.

-infile <cert_file>

Required. Full path to the location of the newly added certificate.

-trustcert

Optional. Checks that the user provider certificate being added is a CA certificate. Smkeytool checks that the certificate has a digital signature extension and that the certificate has the same IssuerDN and Subject DN values.

-noprompt

(Optional) The user will not be prompted to confirm the addition of the certificate.

addRevocationInfo Option

Specifies the location of a CRL so the smkeydatabase can locate the list during the SAML authentication process. The smkeydatabase does not store the contents of a CRL, but merely reads the CRL contents when the Policy Server starts and after a refresh interval has elapsed.

Important! If you add a CRL entry to the smkeydatabase, you must restart the Policy Server.

Arguments for addRevocationInfo are as follows:

-issueralias <issuer_alias>

Required. Alias name of the Certificate Authority who issues the CRL.

Example: -issueralias verisignCA

-type (ldapcrl | filecrl)

Required. Specifies whether the list is a certificate file or an LDAP CRL. The options are ldapcrl or filecrl.

-location <location>

Required. Specifies the location of the CRL. For a file, specify the full path to the file. For an LDAP CRL, specify the full path to the LDAP server node.

Example of file location: -location c:\crls\siteminder_root_ca.crl

Example of LDAP CRL location: -location "http://localhost:880/sn=siteminderroot,dc=crls,dc=com"

changePassword Option

Permits you to change the password for the smkeydatabase. Changing the password causes all entries encrypted under the old password to be re-encrypted under the new password using FIPS-compliant algorithms.

Arguments for changePassword are as follows:

-password <password>

Required. Specifies the existing password used to originally create the smkeydatabase

-newpassword <new_password>

Required. Specifies the new password for the smkeydatabase.

deleteRevocationInfo Option

Deletes a CRL from the database.

Arguments for -deleteRevocationInfo are as follows:

-issueralias <issuer_alias>

Required. Name of the Certificate Authority who issues the CRL.

-noprompt

(Optional) The user will not be prompted to confirm the deletion of the CRL from the database.

deleteDB Option

Deletes the smkeydatabase based on configuration data in the smkeydatabase.properties file. All the entries in the key database and the aliases data store file will be deleted.

Argument for -deleteDB is as follows:

-noprompt

(Optional) The user will not be prompted to confirm the deletion of the database.

delete Option

Deletes an existing certificate from the smkeydatabase. If the certificate has an associated private key, the key is also deleted.

Argument for -delete is as follows:

-alias <alias>

Required. Alias of the certificate to be removed.

-noprompt

(Optional) The user will not be prompted to confirm the deletion of the database.

export Option

Exports an existing certificate or a private key from the smkeydatabase to a file. Certificate data is exported using PEM encoding. Private key data is exported using DER encoded PKCS8 format.

Arguments for the -export option are as follows:

-alias <alias>

Required. Identifies the certificate or key to be exported.

-outfile <out_file>

Required. Specifies the full path to the output file for the exported certificate or key.

-type (key|cert)

Optional. Indicates whether a certificate or key is being exported. If no option is specified, a certificate is the default.

-password <password>

Required only when exporting a private key. Specifies the password used to encrypt the private key at the time the key gets exported to a file. You do not need a password to export the certificate holding the public key because certificates are exported in clear text.

When a private key is exported, it gets exported to the output file in encrypted form using this password. To add this same private key back to the smkeydatabase, run the -addPrivKey command and use this password.

findAlias Option

Determines the alias associated with a certificate that is already in the smkeydatabase.

Argument for -findAlias is as follows:

-infile <cert_file>

Required. Full path to the certificate file associated with the alias you want to find

-password <password>

Password required only when a password-protected P12 file is specified as the certificate file.

importDefaultCACerts Option

Imports all default trusted Certificate Authority certificates from the cacerts.keystore file, which is installed with the Policy Server, into the smkeydatabase. Certificate Authority certificates are used to verify the server certificate associated with the web server at the asserting party.

listCerts Option

Lists some metadata of all the certificates stored in key database.

Argument for -listCerts is as follows:

-alias <alias>

(Optional) Lists the metadata details of the certificate and key associated with the alias specified. This option supports the asterisk (*) as a wildcard character. You can use this wildcard at the beginning and/or at the end of an alias value. Always enclose the asterisk in quotes to avoid a command shell from interpreting the wildcard character.

listRevocationInfo Option

Displays a list of current CRLs in the smkeydatabase. The -listRevocationInfo option only prints the CRL name, type (file or ldap), and the location of all the CRLs in the database.

Argument for -listRevocationInfo is as follows:

-issueralias <issuer_alias>

(Optional) Name of the Certificate Authority who issues the CRL. This option supports the asterisk (*) as a wildcard character. You can use this wildcard at the beginning and/or at the end of an alias value. Always enclose the asterisk in quotes to avoid a command shell from interpreting the wildcard character.

printCert Option

Displays some metadata of the specified certificate. This command is especially useful for UNIX systems, where it is difficult to see the certificate properties.

Arguments for `-printCert` are as follows:

-infile <cert_file>

Required. Location of the certificate file.

-password <password>

Password required only when a password-protected P12 file is specified as the certificate file.

renameAlias Option

Renames an existing alias associated with a certificate.

Arguments for `-renameAlias` are as follows:

-alias <current_alias>

Required. Current alias associated with a certificate.

-newalias <new_alias>

Required. New alias name. Value must be a unique string and should contain only alphanumeric characters.

validateCert Option

(Optional) Indicates whether a certificate is revoked or not.

Arguments for `-validateCert` are as follows:

-alias <alias>

Required. Alias to the certificate associated with this private key in the database. Must be a unique string and should contain only alphanumeric characters.

-infile <crl_file>

Optional. Specifies the CRL file that you want smkeytool to look in for the certificate to validate it.

help Option

Shows how to use the smkeytool utility.

Smkeytool Examples for Windows Platforms

The following are examples of using smkeytool to manage the smkeydatabase.

Example: Create a key database

This example shows the command for creating an smkeydatabase:

```
smkeytool.bat -createDB -password smdb
```

Example: Add a private key/certificate pair

The following example adds a private key/certificate pair to the smkeydatabase. The syntax is the same regardless of whether the key/certificate pair is used for signing and verification or encryption and decryption.

If you run smkeytool from the directory containing the private key/certificate pair, do not specify a directory path in the command line. The command syntax is as follows:

```
smkeytool.bat -addPrivkey -password keypswd -alias privkey1  
-keyfile sampleprivkey.pkcs8" -certfile samplecert.crt"
```

If you run smkeytool from a directory that does not contain the private key/certificate pair, specify the full path to the directory with the pair. The command syntax is as follows:

```
smkeytool.bat -addPrivkey -password keypswd -alias privkey1 -keyfile "c:\program  
files\ca\siteminder\certs\sampleprivkey.pkcs8"  
-certfile "c:\program files\ca\siteminder\certs\samplecert.crt"
```

Example: Add a standalone certificate

The following example adds only a certificate to the smkeydatabase. This certificate can be associated with a private key/certificate pair, but this command only adds the certificate.

If you run the smkeytool from the directory containing the certificate, do not specify a directory path in the command line. The command syntax is as follows:

```
smkeytool.sh -addCert -password keypswd -alias sp2cert -certfile samplefile.crt
```

If you run smkeytool from a directory that does not contain the certificate, specify the full path to directory with the certificate. The command syntax is as follows:

```
smkeytool.sh -addCert -alias sp2cert -certfile  
"export/ca/siteminder/certs/samplefile.crt"
```

Example: Add a trusted CA certificate

The following example shows the commands required to add a trusted Certificate Authority (CA) certificate. For federated communication, SiteMinder can use a trusted CA for securing the back channel for HTTP-Artifact single sign-on.

Important! Obtain a CA certificate from a certificate authority before adding a trusted certificate.

To add a trusted CA certificate

1. Verify whether the certificate exists in the relying party database by entering:
`smkeytool.sh -listCerts`
2. Add the CA certificate by entering:
`smkeytool.bat -addCert "c:\program files\ca\siteminder\certs\sampleCARoot.crt" -trustcert`
3. (Optional) Restart the Policy Server to see the change to the key database immediately.

If you do not restart the Policy Server, it takes some time before the Policy Server and database synchronize. SiteMinder updates the key database based on the value of the `DBUpdateFrequencyMinutes` parameter in the `smkeydatabase.properties` file. You can adjust the frequency by modifying this parameter.

Smkeytool Examples for UNIX Platforms

The following are examples of using smkeytool to manage the smkeydatabase.

Example: Create a key database

This example shows the command for creating the key database:

```
smkeytool.sh -createDB -password siteminderdb
```

Example: Add a private key/certificate pair

The following example adds a private key/certificate pair to the smkeydatabase. The syntax is the same regardless of whether the key/certificate pair is used for signing and verification or encryption and decryption.

If you run smkeytool from the directory containing the private key/certificate pair, do not specify a directory path in the command line. The command syntax is as follows:

```
smkeytool.sh -addPrivkey -password keypswd -alias privkey1 -keyfile privkey.pkcs8  
-certfile sample.crt
```

If you run smkeytool from a directory that does not contain the private key/certificate pair, specify the full path to the directory with the pair. The command syntax is as follows:

```
smkeytool.sh -addPrivkey -alias privkey1 -keyfile "export/ca/siteminder/certs/  
sampleprivkey.pkcs8" -certfile "export/ca/siteminder/certs/sample.crt"
```

Example: Add a standalone certificate

This example command adds only a certificate to the smkeydatabase. This certificate can be associated with a private key/certificate pair, but this command only adds the certificate.

If you run the smkeytool from the directory containing the certificate, do not specify a directory path in the command line. The command syntax is as follows:

```
smkeytool.sh -addCert -password keypswd -alias sp2cert -certfile samplefile.crt
```

If you run smkeytool from the directory that does not contain the certificate, specify the full path to directory with the certificate. The syntax is as follows:

```
smkeytool.sh -addCert -alias sp2cert -certfile  
"export/ca/siteminder/certs/samplefile.crt"
```

Example: Add a trusted CA certificate

This example shows the commands required to add a trusted Certificate Authority (CA) certificate. For federated communication, SiteMinder can use a trusted CA for securing the back channel for HTTP-Artifact single sign-on.

Important! Obtain a CA certificate from a Certificate Authority before adding a trusted certificate.

To add a trusted CA certificate

1. Verify whether the certificate exists in the relying party database by entering:
`smkeytool.sh -listCerts`
2. Add the CA certificate by entering:
`smkeytool.sh -addCert -alias -sp1cacert -infile /opt/netegrity/siteminder/certs/sampleCARoot.cer -trustcacert`
3. (Optional) Restart the Policy Server to see the change to the key database immediately.
4. If you do not restart the Policy Server, it takes some time before the Policy Server and database synchronize. SiteMinder updates the key database based on the value of the DBUpdateFrequencyMinutes parameter in the smkeydatabase.properties file. You can adjust the frequency by modifying this parameter.

Migrate AM.keystore and Update smkeydatabase

Prior to SiteMinder 6.0 SP 5/6.x QMR 5, SiteMinder had the following PKI infrastructure:

- AM.keystore
This store resided on the Web Agent at the relying party. The AM.keystore held Certificate Authority (CA) certificates and client certificates.
- smkeydatabase
This store resided on the asserting and relying party Policy Server systems. The certificates in this key database did not have corresponding aliases.

Beginning with SiteMinder 6.0 SP 5/6.x QMR 5, all data currently stored in the AM.keystore is now stored in the smkeydatabase at the relying party. Additionally, all certificates must have aliases.

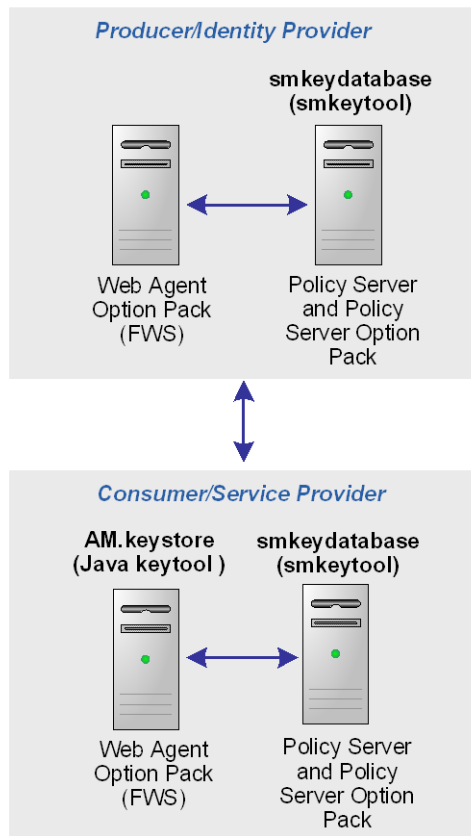
If you are upgrading from versions prior to 6.0 SP 5/6.x QMR 5, you *must* do the following:

1. Copy private keys and certificates from the AM.keystore to the smkeydatabase on the relying party Policy Server.
2. Migrate an existing smkeydatabase so aliases can be added to the certificates in the database.

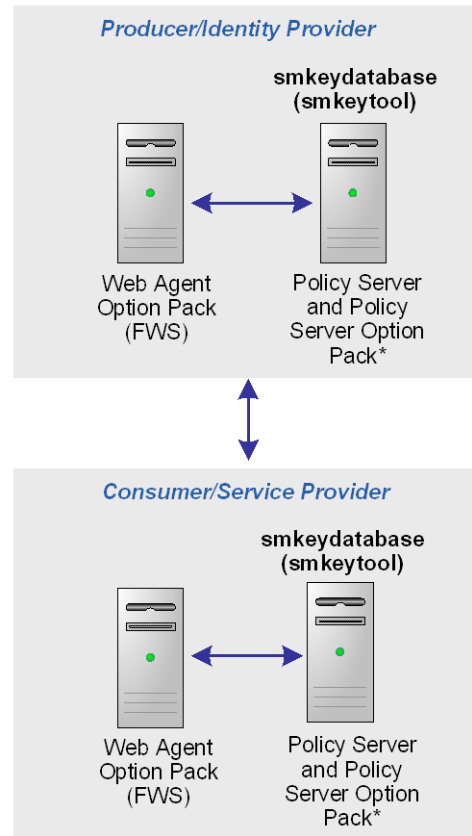
The migratekeystore utility enables you to perform these tasks.

The following picture shows the changes in the PKI infrastructure.

Key Databases Prior to SiteMinder 6.0 SP5



Key Databases Beginning with SiteMinder 6.0 SP5



* Policy Server Option Pack integrated with the Policy Server at r12 SP1

More Information:

[Run the migratekeystore Tool](#) (see page 212)

Considerations Before Migrating Key Databases

Important! Before you migrate the AM.keystore, back up the previous AM.keystore file and the smkeydatabase.properties file. The AM.keystore file is in the location you specified when you first created it.

The smkeydatabase.properties file is in one of the following directories:

Windows: *policy_server_home\config\properties*

UNIX: *policy_server_home/config/properties*

policy_server_home

Indicates the installation directory of the policy server.

Before you migrate, consider the following:

- Understand the purpose of aliases in the smkeydatabase

Aliases enable you to reference any private key/certificate pair in the smkeydatabase. Beginning with 6.0 SP 5/6.x QMR 5, every private key/certificate pair in the smkeydatabase must have a unique alias.

If you are upgrading from a release prior to 6.0 SP5/6.x QMR 5, your existing smkeydatabase must be migrated to the new model using the migratekeystore tool.

When you run the tool the first time, an alias data store is created and aliases are added to this store. For existing private key/certificate entries in the smkeydatabase, an alias is created based on the CN value of the certificate subject DN. If the CN attribute does not exist, the first attribute value of the certificate's subject DN is chosen as the alias. If there are duplicate entries, the alias name is calculated using a combination of multiple attribute values from the subject DN.

Note: You can change an alias value using the renameAlias option of the smkeytool utility.

- Know the Order that SiteMinder Looks for Keys

The Policy Server at the asserting party uses an enterprise private key to sign SAML messages and to decrypt encrypted SAML messages received from the relying party.

When SiteMinder looks for a private key in the database, it searches using the following order of preference:

1. Looks for the key associated with the signing alias, if a signing alias is configured
2. Looks for the default enterprise private key. Typically, the default enterprise key is the first private key found in the smkeydatabase.
3. If there is no default enterprise private key, SiteMinder looks for the first private key in the database.

If you copy data from an AM.keystore to a 6.0 SP6 smkeydatabase, additional private keys and client certificates get added to the smkeydatabase. This may change the order of private keys already in the smkeydatabase. As a result, when you run the migratekeystore tool, it adds an alias named *defaultEnterprisePrivateKey* for the first private key it finds in the database. If a signing alias is not configured, the Policy Server will use the *defaultEnterprisePrivateKey* alias as the key for digital signing.

More Information

[Run the migratekeystore Tool](#) (see page 212)

How To Migrate the Key Databases

Use the migratekeystore tool to update your pre 6.0 SP5/6.x QMR 5 PKI infrastructure.

1. Do the following prior to running the migratekeystore tool:
 - Copy the AM.keystore file from the Web Agent machine to the Policy Server with the smkeydatabase.
 - Gather any passwords associated with client certificates that were in the AM.keystore. You will need to specify these passwords during the migration.
2. Run the migratekeytool utility.

Run the migratekeystore Tool

This procedure accomplishes two tasks:

- migrating an AM.keystore to an smkeydatabase
- updating an existing smkeydatabase so certificates have aliases.

Note: If you have a clustered Policy Server environment, perform this procedure one time on one system then copy the entire smkeydatabase directory to the other machines in the cluster.

To migrate the AM.keystore and update existing smkeydatabase certificates

1. Back up your existing databases.
2. Open a command window.
3. Copy the AM.keystore file from the machine where the Web Agent Option Pack is installed and place the file on the machine with the Policy Server installed.

Important! If you are only updating certificates in an existing smkeydatabase, skip to Step 4.

The location of the AM.keystore is:

web_agent_home/affwebservices/AM.keystore

Copy the file to:

policy_server_home/siteminder/smkeydatabase

If the smkeydatabase does not exist, create a database using the smkeytool -createDatabase command.

4. Enter one of the following commands to complete the migration and update:

Windows:

```
migratekeystore.bat java_keystore_location java_keystore_password
```

UNIX:

```
migratekeystore.sh java_keystore_location java_keystore_password
```

java_keystore_location

location of the am.keystore file

java_keystore_password

password to access the contents of the am.keystore file. Passwords are shown in clear text.

As the tool processes the command, you are prompted to answer a series of questions about the data you want to copy. After answering the questions, the data is copied and the smkeydatabase is updated.

Note: Any migrated data will be encrypted using FIPS-compliant algorithms.

Chapter 10: Securing a Federated Environment

This section contains the following topics:

[Protecting Federated Communication](#) (see page 215)

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 SiteMinder, here are some recommendations for protecting your environment:

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

These topics are described in the sections that follow.

Setting a One Time Use Condition for an Assertion

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

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

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

Securing Connections Across the Federated Environment

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

Securing the Connection Between the Relying Party and the Target Application

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

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

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

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

Securing the Initial Authentication at the SiteMinder Asserting Party

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

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

Protecting Against Cross-Site Scripting

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

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

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

- `idpdiscovery.jsp`
Used at the relying party for Identity Provider Discovery.
- `linkaccount.jsp`
Used at the relying party for dynamic account linking.
- `sample_application.jsp`
Used at the IDP to initiate single sign-on. You can use this sample application to direct the user to the SSO Service and then to the custom web application. Typically, you use your own application.
- `signoutconfirmurl.jsp`
Used at the Account Partner for WS-Federation sign out.
- `unsolicited_application.jsp`
Used for IdP-initiated single sign-on when the user is sent directly to the web application and not initially to the SSO Service.

The pages scan the request for the following characters:

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

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

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

Chapter 11: Creating Affiliate Domains

This section contains the following topics:

[Affiliate Domain Overview](#) (see page 219)

[Configure an Affiliate Domain](#) (see page 219)

Affiliate Domain Overview

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

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

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

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

More Information:

[Assign User Directories](#) (see page 220)

Configure an Affiliate Domain

An affiliate domain contains entities associated with one or more user directories. Affiliate domains require an association with one or more administrator accounts that can make changes to the objects in the domain.

To configure an affiliate domain and add entities to the domain:

1. [Add a Domain Object](#) (see page 220)
2. [Assign User Directories](#) (see page 220)

3. [Assign an Administrator](#) (see page 221)
4. [Add Entities to an Affiliate Domain](#) (see page 222)

Add a Domain Object

To configure an affiliate domain

1. Log into the FSS Administrative UI.
2. From the menu bar of the SiteMinder Administration window, select Edit, System Configuration, Create Domain.

The SiteMinder Domain dialog box opens.

3. In the Domain Type group box, select the Affiliate Domain radio button.
The tabs in the dialog box allow you to enter information for an affiliate domain.
4. In the Name field, enter a name for the affiliate domain.
5. In the Description field, enter a brief description of the affiliate domain.

Assign User Directories

Select users that should have access to resources at the consumer, Service Provider, or Resource Partner.

In the User Directories tab of the Domain Properties dialog box, specify the user directories that contain the users who should be authenticated and authorized for access to the affiliate resources.

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

To assign user directories to an affiliate domain:

1. Select the User Directories tab.
2. From the drop-down list box at the bottom of the tab, select a user directory you want to include in the affiliate domain.
3. Click the Add button.

The FSS Administrative UI adds the directory to the list displayed in the User Directories tab.

For user directories that serve as the authentication directory in a directory mapping, the list displays the authorization directory and the method of directory mapping.

4. Repeat steps 2 and 3 for all user directories you want to associate with the domain.

Note: The order in which you add directories to the domain is the order in which SiteMinder searches to find user records, starting from the top of the list. You can use the arrow buttons to the right of the list of directories to change the order of directories.

5. Optionally, you can create a user directory from this dialog box by clicking Create toward the bottom of the tab.

The User Directory dialog box opens. Create the user directory. When you save the new User Directory and close the User Directory dialog box, the directory you created appears in the User Directories tab in the Domain Properties dialog box.

6. Click OK to save your changes.

Assign an Administrator

When you create an affiliate domain, you can assign administrators to the domain. These administrators may create, edit, and delete entities within the domain.

To assign an Administrator to an affiliate domain

1. Select the Administrators tab.
2. From the drop-down list box at the bottom of the tab, select an administrator.
3. Click Add.

The administrator is added to the list in the top portion of the Administrators tab. When you save the affiliate domain, administrators included in the list can manage objects within the affiliate domain.

Note: Adding an administrator in the Administrators tab is equivalent to adding the affiliate domain to an administrator with a Scope of Some Domains and a task of Manage Domain Objects.

4. Optionally, create an administrator by clicking Create at the bottom of the tab.

The Administrator dialog box opens. When you save the new administrator and close the Administrator Properties dialog box, the FSS Administrative UI displays the administrator in the Administrators tab.

5. Click OK to save your changes.

Add Entities to an Affiliate Domain

You can add the following consuming authorities to an affiliate domain:

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

Note: These entities must be given permission to access Federation Web Services at the asserting party when you protect the Federation Web Services application.

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

- To add a SAML 1.x consumer, review the instructions for identifying consumers for a SAML 1.x producer.
- To add a SAML 2.0 Service Provider, review the instructions for identifying Service Providers for a SAML 2.0 Identity Provider.
- To add a WS-Federation Resource Partner, review the instructions for identifying Resource Partners at the Account Partner.

More information:

[Configure SiteMinder as a SAML 1.x Producer](#) (see page 223)

[Configure SiteMinder as a SAML 2.0 Identity Provider](#) (see page 285)

[Configure SiteMinder as an Account Partner](#) (see page 401)

Chapter 12: Configure SiteMinder as a SAML 1.x Producer

This section contains the following topics:

- [Prerequisites for a SiteMinder Asserting Party](#) (see page 223)
- [How To Configure SiteMinder to Act as a SAML 1.x Producer](#) (see page 224)
- [Add a Consumer to an Affiliate Domain](#) (see page 225)
- [Authenticate Users with No SiteMinder Session \(SAML 1.x\)](#) (see page 227)
- [Select Users for Which the Producer Generates Assertions](#) (see page 228)
- [Configure a SAML 1.x Assertion](#) (see page 231)
- [Grant Access to the Service for Assertion Retrieval \(Artifact SSO\)](#) (see page 234)
- [Configure the Authentication Scheme that Protects the Artifact Service](#) (see page 236)
- [Setting Up Sessions for a SAML Affiliate Agent Consumer \(optional\)](#) (see page 241)
- [Configure Attributes to Include in SAML 1.x Assertions \(Optional\)](#) (see page 243)
- [Configure IP Address Restrictions for 1.x Consumers \(optional\)](#) (see page 248)
- [Configure Time Restrictions for 1.x Consumers \(optional\)](#) (see page 248)
- [Customize the SAML 1.x Assertion Response \(optional\)](#) (see page 249)
- [Creating Links to Consumer Resources for Single Sign-on](#) (see page 251)

Prerequisites for a SiteMinder Asserting Party

For SiteMinder to serve as the asserting party, the following conditions must be met:

- Install the Policy Server.
- Install one of the following:
 - The Web Agent and the Web Agent Option Pack. The Web Agent authenticates users and establishes a SiteMinder session. The Option Pack provides the Federation Web Services application.
 - The SPS federation gateway, which has an embedded Web Agent and has the Federation Web Services application deployed on the embedded Tomcat web server.
- A SAML consumer is set up within the federated network.

The assertions generated by the producer are sent to the consumer. An application at the consumer receives and interprets the assertion. The SAML Affiliate Agent and a system with the SiteMinder Web Agent Option Pack can act as SAML consumers.

How To Configure SiteMinder to Act as a SAML 1.x Producer

The producer requires information about the consumer to send s SAML response for a given request.

Note: The SAML 1.x consumer is named an affiliate in the FSS Administrative UI.

The following configuration tasks at the producer are required:

1. Associate the affiliate with an affiliate domain. The affiliate must have permission to access Federation Web Services at the Producer.
2. Provide general information about the affiliate.
3. Select the users for which the producer generates assertions.
4. Configure the assertions.
5. The session server, a component of the Policy Server, is enabled. For SAML artifact authentication, the session server is where assertions are stored before they are forwarded to the Federation Web Services application at the Consumer.

The following configuration tasks are optional:

- Configure session management when the SAML Affiliate Agent is acting as the consumer.
- Configure attributes for inclusion in the assertions.
- Set IP address restrictions to limit the addresses used to access the affiliate.
- Configure time restrictions for affiliate operation.
- Configure the Assertion Generator plug-in to customize the content of an assertion.

Tips:

- Certain parameter values at the Producer and Consumer are required to match for the configuration to work. A list of those parameters is in [Configuration Settings that Must Use the Same Values](#) (see page 471).
- Verify that you are using the correct URLs for the Federation Web Services servlets. The URLs are listed in [Federation Web Services URLs Used in SiteMinder Configuration](#) (see page 477).

Optional Configuration Tasks at a 1.x Producer

The following are optional tasks for identifying a consumer at the producer site:

- If the SAML Affiliate Agent is acting as the consumer, configure session management.
- Configure attributes for inclusion in assertions.
- Set IP address restrictions to limit the addresses used to access consumers.
- Configure time restrictions for consumer operation.
- Configure the Assertion Generator plug-in to customize assertion content.

More Information:

[Setting Up Sessions for a SAML Affiliate Agent Consumer \(optional\)](#) (see page 241)

[Configure Attributes to Include in SAML 1.x Assertions \(Optional\)](#) (see page 243)

[Configure IP Address Restrictions for 1.x Consumers \(optional\)](#) (see page 248)

[Configure Time Restrictions for 1.x Consumers \(optional\)](#) (see page 248)

[Customize the SAML 1.x Assertion Response \(optional\)](#) (see page 249)

Add a Consumer to an Affiliate Domain

Entities that consume SAML 1.x assertions are called consumers in the Federation Security Services documentation. However, in the Policy Server User Interface, the term affiliate is used to represent the consumer. When used in the Policy Server User Interface, the term affiliate is synonymous with consumer.

To add a consumer to an affiliate domain

1. Log into the FSS Administrative UI.
2. Display the list of domains.
3. Expand the affiliate domain where you want to add a consumer.
4. Click on the Affiliates icon.
5. From the menu bar, select Edit, Create Affiliate.

The Affiliate dialog box opens.

6. Complete the following required fields.

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

- Name
- Password and Confirm Password (For SAML artifact only)
- Authentication URL

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

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

myserver

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

Note: You will need to create a policy to protect the AuthenticationURL.

7. Select the Enabled check box to activate the affiliate object.

This check box must be marked for the Policy Server and Federation Web Services to support authentication for the consumer resources.

8. Optionally, check the Use Secure URL check box.

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

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

`http(s)://idp_server:port/affwebservices/secure/securedirect.`

Click Help for more details about this field.

9. Optionally, if the SAML Affiliate Agent is acting as the SAML consumer, select the Allow Notification check box to provide event notification services for the consumer.

The notification feature allows the producer to track user activity at the consumer. If this check box is selected, the producer can receive event notifications from the consumer about which resources a user has accessed. When the user accesses specific URLs at the consumer, the consumer may notify the producer. The producer can log this activity and use the information for auditing or reporting purposes.

Important! The Notification service is not supported with the SAML credential collector acting as a consumer.

More Information:

[Authenticate Users with No SiteMinder Session \(SAML 1.x\)](#) (see page 227)

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

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

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

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

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

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

Create a Policy to Protect the Authentication URL

To create a policy to protect the AuthenticationURL

1. Open the FSS Administrative UI.
2. From the System tab, create Web Agents to bind to the realms that you define for the producer-side Web Server. You can assign unique Agent names for the Web Server and the Federation Web Services or use the same Agent name for both.
3. Create a policy domain for the users who should be challenged when they try to access a consumer resource.
4. From the Users tab, select the users that should have access to the resources that are part of the policy domain.

5. Define a realm for the policy domain with the following values:
 - a. Agent: select the Agent for the producer Web Server
 - b. Resource Filter:

Web Agents v5.x QMR 4 and later, and SPS federation gateway enter:
`/siteminderagent/redirectjsp/`

Web Agents v5.x QMR 1, 2, or 3, enter:
`/affwebservices/redirectjsp/`

The resource filter `/siteminderagent/redirectjsp/` is an alias, set up automatically by FWS. It is a reference to the following:
 - For a Web Agent:
`web_agent_home/affwebservices/redirectjsp`
 - For the SPS federation gateway:
`sps_home/secure-proxy/Tomcat/webapps/affwebservices/redirectjsp`
 - c. For the remaining settings, accept the defaults or modify as needed.
6. For SAML artifact only, select the Session tab and check the Persistent Session check box.

To enable single sign-on using the SAML artifact profile from a realm at the producer to a realm at the consumer, configure a persistent session for the producer realm. If you do not configure a persistent session, the user cannot access consumer resources.
7. Click OK to save the realm.
8. Create a rule for the realm. In the Resource field, accept the default value, the asterisk (*), to protect all resources for the realm.
9. Create a policy for the producer Web Server that includes the rule created in the previous step.
10. Complete the task in [Select Users for Which Assertions Will Be Generated](#) (see page 228).

Select Users for Which the Producer Generates Assertions

The next step in defining a consumer for the producer is to include a list of users and groups for which the assertion generator will generate assertions. The users need assertions to authenticate at the consumer site. You may only add users and groups from directories included in an affiliate domain.

Adding Users and Groups for Access to a Consumer

You define which users and groups the assertion generator creates assertions for and include those users as part of the consumer's configuration.

To specify which users can obtain assertions

1. In the SiteMinder Affiliate dialog box, click on the Users tab.

If the associated affiliate domain contains more than one user directory, the directories appear as tabs on the Users tab.

2. Click the Add/Remove button.

The Users/Groups dialog box opens.

3. To add users, select an entry from the Available Members list and click the Left Arrow button, which points to the Current Members list.

The opposite procedure removes users from the Current Members list.

You can select multiple entries by holding the CTRL or SHIFT key and clicking entries in one of the Members lists. When you select multiple entries and click one of the Arrow buttons, the FSS Administrative UI moves all of the selected entries.

Individual users are not displayed automatically. However, you can use the Search utility to find a specific user in one of the listed groups. Different types of user directories must be searched differently.

4. Click OK to save your changes.

Excluding a User or Group from Access to a Consumer

You can exclude users or groups of users from obtaining an assertion. This is useful if you have a large user group that should have access to a consumer, but you there is a small subset of this group that you want to exclude.

To exclude a user or group from gaining access to a consumer's resources:

1. In the Users/Groups dialog box, select a user or group from the Current Members list.

2. Click Exclude to exclude the selected user or group.

The symbol to the left of the user or group in the Current Members list changes to indicate that the user or group is excluded from the consumer.

When you exclude a group from resource access, the assertion generator will not create an assertion for anyone who is a member of the excluded group.

3. Click OK to save your changes.

Allowing Nested Groups Access to Consumers

LDAP user directories may contain groups that contain sub-groups. In complex directories, groups nesting in a hierarchy of other groups is one way to organize tremendous amounts of user information.

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

To allow nested groups from within an LDAP directory

1. In the Affiliate Properties dialog box, click on the Users tab.

If the associated affiliate domain contains more than one user directory, the directories appear as tabs on the User tab.

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

Adding Users by Manual Entry

From the Users/Groups dialog box, you can use the Manual Entry option to add users who should have access to a consumer.

To add a user by manual entry

1. In the Manual Entry group box, do one of the following:
 - For LDAP directories, enter a valid DN in the Entry field. For each DN specified in the Entry field, you can select an action from the Action drop down list, as follows:
 - Search Users--the LDAP search is limited to matches in user entries.
 - Search Groups--the LDAP search is limited to matches in group entries.
 - Search Organizations--the LDAP search is limited to matches in organization entries.
 - Search Any Entry--the LDAP search is limited to matches in user, group, and organization entries.
 - Validate DN--the LDAP search locates this DN in the directory.

- For Microsoft SQL Server, Oracle and WinNT directories, enter a user name in the Manual Entry field.

For an Microsoft SQL Server or Oracle, you can enter a SQL query, instead. For example:

```
SELECT NAME FROM EMPLOYEE WHERE JOB ='MGR';
```

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

Note: For an LDAP directory, you can enter all in the Manual Entry field to add all directory entries to the consumer.

2. Click Add to Current Members.

The FSS Administrative UI adds the user or query to the Current Members list.

3. Click OK to save your changes and return to the SiteMinder Affiliate dialog.

Configure a SAML 1.x Assertion

The Assertions tab lets you define how assertions are sent to the consumer. The assertion is used to authentication the user at the consumer site.

To configure a SAML 1.x assertion

1. Log into the FSS Administrative UI.
2. Click on the Domains tab and select the affiliate domain.
3. Select Affiliates to display the list of consumers, and double-click the consumer you want to configure.

The Affiliate Properties dialog box opens.

4. Click the Assertions tab.

5. Complete the following fields.

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

- SAML Profile
- Assertion Consumer URL (required for SAML POST; optional for SAML Artifact)

For details and required URL syntax for this field, click Help.

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

- Validity Duration Seconds
- Skew Time Seconds

6. Optionally, fill in the Audience field.
7. Optionally, for artifact profile, check the Sign Assertion box.
8. Click OK to save your changes.

More Information:

[Assertion Validity for Single Sign-on](#) (see page 232)

A Security Issue Regarding SAML 1.x Assertions

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

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

Assertion Validity for Single Sign-on

Based on the values of the Validity Duration and Skew Time, the assertion generator calculates the total time that the assertion is valid. In the assertion document, the NotBefore and NotOnOrAfter values represent the beginning and end of the validity interval.

To determine the beginning of the validity interval, the assertion generator takes the system time when the assertion is generated. The assertion generator sets the IssueInstant value in the assertion according to this time. The assertion generator subtracts the Skew Time value from the IssueInstant value. The resulting time becomes the NotBefore value.

To determine the end of the validity interval, the assertion generator adds the Validity Duration value and the Skew Time together. The resulting time becomes the NotOnOrAfter value.

For example, an assertion is generated at the producer 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.

Note: Times are relative to GMT.

Generate an Assertion for One Time Use

You can configure SiteMinder to generate an assertion that includes a one time use condition. If the relying party knows to use the assertion immediately and not cache it for future use, the relying party will not make authentication decisions based on an out-of-date assertion.

To generate an assertion with a one time use condition

1. Log on to the FSS Administrative UI.
2. Select the affiliate you want to modify or create an affiliate.
3. Navigate to the Advanced tab for the affiliate.
4. Select the Set DoNotCache Condition check box.
5. Click OK.

The asserting party can now generate an assertion that includes the condition element for its one time use.

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

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

To grant access:

- [Add the Web Agent](#) (see page 302) that protects the FWS application to the agent group FederationWebServicesAgentGroup.
- [Add relying partners as users](#) (see page 303) who are permitted to access the specific service.

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

Add a Web Agent to the Federation Agent Group

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

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

Follow these steps:

1. Log in to the FSS Administrative UI.
2. Begin at the System tab and select Edit, Create Agent.
The Agent Properties dialog opens.
3. Enter the name of the Web Agent for your deployment. Click OK.
4. Select View, Agent Groups.
5. Select Agent Groups.
6. Select the FederationWebServicesAgentGroup entry.
The Agent Groups dialog opens.
7. Click Add/Remove and the Agent Group Members dialog opens.
8. Move the web agent from the Available Members list to the Selected Members list.
9. Click OK to return to the Agent Groups dialog.
10. Click OK.

Add Relying Partners to the FWS Policy for Obtaining Assertions (Artifact SSO)

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

When you install the Policy Server, the FederationWebServicesDomain is installed by default.

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

Grant access for these policies to any relevant relying partners.

Follow these steps:

1. In the FSS Administrative UI, select the Domains tab.
2. Expand the FederationWebServicesDomain and select Policies.

A list of federation policies displays.

3. Double-click the policy for the appropriate SAML profile:

SAML 1.x

FederationWSAssertionRetrievalServicePolicy

SAML 2.0

SAML2FWSArtifactResolutionServicePolicy

The Policy Properties page opens.

4. From the Users tab, select the tab for the appropriate directory.

SAML 1.x

FederationWSCustomUserStore

SAML 2.0

SAML2FederationCustomUserStore

5. Click Add/Remove.

The existing affiliate domains are listed in the Users/Groups dialog. For example, if the affiliate domain is named fedpartners, the entry is **affiliate:fedpartners**.

6. Select the affiliate domain and move it to the Current Members list.
7. Click OK until you return to the FWS policies list.

You have now added affiliates to an FWS policy.

Verify Basic Protection of the Assertion Retrieval Service

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

Follow these steps:

1. Open a web browser.

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

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

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

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

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

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

Configure the Authentication Scheme that Protects the Artifact Service

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

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

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

You can select one of the following authentication schemes:

- [Basic](#) (see page 315)
- Basic over SSL
- [X.509 client certificate](#) (see page 238)

Basic Authentication to Protect the Service that Retrieves Assertions

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

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

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

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

1. Log in to the FSS Administrative UI.
2. Double-click the affiliate you want to edit.

The Affiliate Properties dialog opens.

3. Enter a value for the following fields:
 - Password
 - Confirm Password
4. Click OK to save the changes.

Follow these steps: for the SAML 2.0 Artifact Resolution Service

1. Log in to the FSS Administrative UI.
2. Double-click the Service Provider you want to edit.
3. From the General tab, select Configure Backchannel Authentication.
4. In the Backchannel Properties dialog, enter a value for the following fields:
 - Password
 - Confirm Password
5. Click OK to save the changes.

Basic over SSL to Protect the Assertion Retrieval Service

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

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

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

Client Certificate Auth to Protect the Service that Retrieves Assertion

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

To use a client certificate authentication scheme:

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

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

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

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

More Information:

[Configure the Client Certificate Authentication at the Relying Party](#) (see page 381)

Create the Policy to Protect the Retrieval Service

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

Follow these steps:

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

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

```
uid=CompanyA, ou=Development, o=CA
```

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

2. Add the configured user directory to the FederationWebServicesDomain.
3. Create a certificate mapping entry.

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

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

4. Configure an X509 Client Certificate authentication scheme.
5. Create a realm under the FederationWebServicesDomain containing the following entries:

Name

any_name

Example: cert assertion retrieval

Agent

FederationWebServicesAgentGroup

Resource Filter

/affwebservices/certassertionretriever (SAML 1.x)

/affwebservices/saml2certartifactresolution (SAML 2.0)

Authentication Scheme

Client certificate authentication scheme created in the previous step.

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

Name

any_name

Example: cert assertion retrieval rule

Resource

*

Web Agent Actions

GET, POST, PUT

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

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

Create a response with the following values:

Name

any_name

Attribute

WebAgent-HTTP-Header-Variable

Attribute Kind

User Attribute

Variable Name

consumer_name

Attribute Name

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

Example: uid=CompanyA.

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

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

Name

any_name

User

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

Rule

rule_created_earlier_in_this_procedure

Response

response_created_earlier_in_this_procedure

The policy to protect the artifact resolution service is complete.

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

SAML 1.x: [Enable client certificate authentication](#) (see page 275) for the Assertion Retrieval Service

SAML 2.0: [Enable client certificate authentication](#) (see page 275) for the Artifact Resolution Service

Setting Up Sessions for a SAML Affiliate Agent Consumer (optional)

To configure sessions for a site using the SAML Affiliate Agent as the consumer, be aware of the following information:

- Session management is configured only if the SAML Affiliate Agent is acting as a SAML consumer. The SAML credential collector does not support this feature.
- The SAML Affiliate Agent does not support the SAML POST profile. Sessions can only be used with the SAML artifact profile.

Session management between a producer and a SAML Affiliate Agent can be handled in one of three ways:

Default

Producer and SAML Affiliate Agent maintain separate sessions.

Both the producer and the SAML Affiliate Agent establish sessions for the user. If a user idles out or the user reaches a timeout at the producer, the SAML Affiliate Agent is not notified. The same is true for a session that expires at the SAML Affiliate Agent.

Active

An active session is required at the producer.

Both the producer and SAML Affiliate Agent establish sessions. An active session is required at the producer for the SAML Affiliate Agent session to stay active. Producer sessions can remain active after a SAML Affiliate Agent session is terminated.

Shared

Producer and SAML Affiliate Agent maintain shared sessions.

If the SAML Affiliate Agent has implemented a shared-session model, the producer and the SAML Affiliate Agent can maintain a shared session. If the producer session expires or the user logs out at either site, the producer or the SAML Affiliate Agent terminate the sessions.

Note: For more information about session management, see the *SAML Affiliate Agent Guide*.

Configure a Default or Active Session Model

For the Default and Active session models, no specific configuration is required at the Producer. The configuration takes place at the SAML Affiliate Agent.

Configure a Shared Session Model

For shared sessioning, there are a few steps to complete. You configure shared sessioning on the Session tab of the Affiliate Properties dialog box.

To enable shared sessioning

1. Select the Shared Sessioning checkbox.

If a SAML Affiliate Agent implements a shared session solution, this check box enables the sharing of session information between the producer and the SAML Affiliate Agent.

2. Enter a value in the Sync Interval field, in seconds.
3. Click OK to save your changes.

More Information:

[Set the Sync Interval for Shared Sessions](#) (see page 242)

Set the Sync Interval for Shared Sessions

The sync interval defines the frequency at which the SAML Affiliate Agent contacts the producer to validate session status. The SAML Affiliate Agent learns the value of the sync interval from the assertion.

The sync interval helps ensure that the information at the session store and the information in the SAML Affiliate Agent is synchronized. For example, imagine that the sync interval is 2 minutes, and the user logs out at the producer at 4:00PM. The consumer session cookies do not become invalid until 4:02PM.

Note: The SAML Affiliate Agent does not automatically contact the producer only because of the value of the sync interval. The user has to be active at the consumer--that is, the user is requesting consumer resources.

Two factors affect the value of Sync Interval:

- If the value of Sync Interval is too small, the SAML Affiliate Agent keeps calling the session store. The continual calls slow down performance.
- The value must not be larger than 1/2 the lowest Idle Timeout Enabled value that is set for the realm where the user logs in. The Idle Timeout Enabled field is part of the session configuration for the realm.

Note: If the user visits the SAML Affiliate Agent *before* logging in at the producer, the user is redirected to a URL at the producer. This URL is referred to as the PortalQueryURL.

Configure Attributes to Include in SAML 1.x Assertions (Optional)

You can include attributes in assertions. Servlets or applications can use attributes to display customized content for a user. User attributes, DN attributes, or static data can all be passed from the producer to the consumer in an assertion. When used with web applications, attributes can limit the activities of a user at the consumer. For example, the producer sends an attribute named Authorized Amount. The consumer sets this attribute to a maximum dollar amount that the user can spend.

Attributes take the form of name/value pairs and include information, such as a mailing address, business title, or an approved spending limit for transactions. When the consumer receives the assertion, it extracts the attributes. The consumer makes the attributes available to applications as HTTP header variables or HTTP cookie variables.

To pass the attributes, configure a response. The responses available for this purpose are:

- Affiliate-HTTP-Header-Variable
- Affiliate-HTTP-Cookie-Variable

The HTTP headers and HTTP cookies have size restrictions that assertion attributes cannot exceed. The size restrictions are as follows:

- For HTTP headers, SiteMinder can send an attribute in a header up to the web server size limit for a header. Only one assertion attribute per header is allowed. See the documentation for your web server to determine the header size limit.
- For HTTP cookies, SiteMinder can send a cookie up to the size limit for a cookie. Each assertion attribute is sent as its own cookie. The cookie size limit is browser-specific, and that limit is for all attributes being passed to the application, not for each attribute. See the documentation for your web browser to determine the cookie size limit.

More Information:

[Attribute Types](#) (see page 244)

Attribute Types

Attributes identify information that the producer passes to the consumer.

You can specify the following types of attributes as an addition to an assertion:

- Static attribute

Returns data that remains constant.

Use a static attribute to return a string as part of a SiteMinder response. This type of response can be used to provide information to a web application. For example, if a group of users has customized content on a website, the static response attribute, `show_button = yes`, can be passed to the application.

- User attribute

Returns profile information from an entry in a user directory.

This type of response attribute returns information associated with a user in a directory. A user attribute can be retrieved from an LDAP, WinNT, or ODBC user directory.

Note: For the producer to return response attributes with values from user directory attributes, the user directories must be configured in the Administrative UI.

- DN attribute

Returns profile information from a directory object in an LDAP or ODBC user directory.

This type of attribute is used to return information associated with directory objects to which the user is related. Groups to which a user belongs, and Organizational Units (OUs) that are part of a user DN, are examples of DN attributes.

For example, you can use a DN attribute to return a company division for a user, based on the user membership in a division.

Note: For the Policy Server to return response attributes using values from DN attributes, the user directories must be configured in the SiteMinder User Directory dialog.

Configure Attributes for SAML 1.x Assertions

You can configure responses to pass attributes from a SAML assertion to a target application at the consumer site.

To configure an attribute for an assertion

1. In the Affiliate Properties dialog, select the Attributes tab.
2. Click Create.

The Affiliate Attribute Editor dialog opens.

3. From the Attribute drop-down list, select whether you want to configure a header or cookie variable.
4. From the Attribute Setup tab, select one of the following options in the Attribute Kind group box:
 - Static
 - User Attribute
 - DN Attribute

If you select the DN Attribute, you can also select the Allow Nested Groups check box. Selecting this check box allows SiteMinder to return an attribute from a group that is nested in another group specified by a policy. Nested groups often occur in complex LDAP deployments.

Your selection from the Attribute drop-down list and the response attribute type you select determine the available fields in the Attribute Fields group box.

5. Complete the fields for the Attribute Kind you select. The Attribute Kind that you select determines which additional fields you must configure.

Static

Fill in the following fields:

- Variable Name
Enter the name for the attribute SiteMinder returns to the affiliate.
- Variable Value
Enter the static text as the value for the name/value pair.
For example, to return the name/value pair show_content=yes, enter show_content as the variable name and yes as the variable value.

User Attribute

Fill in the following fields:

- Variable Name
Enter the name for the attribute SiteMinder returns to the consumer.
- Attribute Name
Enter the attribute in the user directory for the name/value pair.
For example, to return the email address of a user to the consumer, enter email_address as the Variable Name, and email as the Attribute Name.

DN Attribute

Fill in the following fields:

- Variable Name
Enter the name for the attribute SiteMinder returns to the consumer.
- DN Spec
Enter the distinguished name of the user group from which SiteMinder retrieves the user attribute. The DN must be related to the users for whom you want to return values to the consumer. If you do not know the DN, click Lookup. Use the SiteMinder User Lookup dialog to locate the user group and select a DN.
- Attribute Name
Enter the attribute in the user directory for this attribute for the name/value pair.

Note: If you selected Affiliate-HTTP-Cookie-Variable from the Attribute menu, the Variable Name field label changes to Cookie Name.

6. (Optional) if the LDAP user directory contains nested groups, and you want the Policy Server to retrieve DN attributes from the nested groups, select the Allow Nested Groups check box in the Attribute Kind group box.
7. Click OK to save your changes.

Specify the Maximum Length of Assertion Attributes

The maximum length for user assertion attributes is configurable. To modify the maximum length of assertion attributes, change the settings in the EntitlementGenerator.properties file.

Note: The property name in the file is specific to the protocol you are configuring.

Follow these steps:

1. On the system where the Policy Server is installed, navigate to *policy_server_home*\config\properties\EntitlementGenerator.properties.
2. Open the file in a text editor.
3. Adjust the maximum user attribute length for the protocols in use in your environment. The settings for each protocol are as follows:

WS-Federation

Property Name:

com.netegrity.assertiongenerator.wsfed.MaxUserAttributeLength

Property Type: Positive Integer value

Default Value: 1024

Description: Indicates the maximum attribute length for WS-FED assertion attributes.

SAML 1.x

Property Name:

com.netegrity.assertiongenerator.saml1.MaxUserAttributeLength

Property Type: Positive Integer value

Default Value: 1024

Description: Indicates the maximum attribute length for SAML1.1 assertion attributes.

SAML 2.0

Property Name:

com.netegrity.assertiongenerator.saml2.MaxUserAttributeLength

Property Type: Positive Integer value

Default Value: 1024

Description: Indicates the maximum attribute length for SAML2.0 assertion attributes

4. Restart the Policy Server after any change to these parameters.

Use a Script to Create A New Response Attribute

The Advanced tab of the Affiliate Attribute dialog box contains the Script field. This field displays the script that SiteMinder generates based on your entries in the Attribute Setup tab. You can copy the contents of this field and paste them into the Script field for another response attribute.

Note: If you copy and paste the contents of the Script field for another attribute, you must select the appropriate radio button in the Attribute Kind group box of the Attribute Setup tab.

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

The FSS Administrative UI allows you to specify a single IP address (by address or host name), a range of IP addresses, or a subnet mask that users must use to access a consumer site. The Consumer accepts only those users who access the consumer site from the appropriate IP address.

The procedure for specifying IP address restrictions for a consumer is the same as configuring IP restrictions for a policy in a policy domain.

Configure Time Restrictions for 1.x Consumers (optional)

The FSS Administrative UI lets you to add time restrictions for accessing consumer resources. When you specify a time restriction, the consumer functions only during the period specified in the time restriction. If a user attempts to access a consumer resource outside of the period specified by the time restriction, the producer does not generate SAML assertions.

The procedure for specifying time restrictions for a consumer is the same as specifying them for a policy in a policy domain.

Customize the SAML 1.x Assertion Response (optional)

The SiteMinder Assertion Generator produces SAML assertions to authenticate users in a federation environment. You can customize the content of the SAML assertion generated by the Assertion Generator by configuring an Assertion Generator plug-in. Using this plug-in, you can modify the assertion content for your business agreements between partners and vendors.

To use the Assertion Generator plug-in

1. Implement the plug-in class.

A sample class, `AssertionSample.java`, can be found in `sdk/samples/assertiongeneratorplugin`.

2. Configure the Assertion Generator plug-in from the Advanced tab of the Affiliate Properties dialog.

Note: Specify an Assertion Generator plug-in for each consumer.

- a. In the Full Java Class Name field, enter the Java class name of the plug-in.

For example, `com.mycompany.assertiongenerator.AssertionSample`

A sample plug-in is included in the SDK. You can view a sample assertion plug-in at `sdk/samples/assertiongeneratorplugin`.

- b. Optionally, in the Parameters field, enter the string that gets passed to the plug-in as a parameter at run time.

The string can contain any value; there is no specific syntax to follow.

For more information about the Assertion Generator plug-in (method signatures, parameters, return values, data types), and also the new constructor for `UserContext` class, see the `AssertionGeneratorPlugin` interface in the *Javadoc Reference*. For overview and conceptual information, see the *SiteMinder Programming Guide for Java*.

Implement the AssertionGeneratorPlugin Interface

The first step in creating a custom assertion generator plug-in is to implement the `AssertionGeneratorPlugin` interface.

Follow these steps:

1. Provide a public default constructor method that contains no parameters.
2. Provide code so that the implementation is stateless. Many threads must be able to use a single plug-in class.

3. Implement methods in the interface to satisfy your requirements.

The implementation must include a call to the `customizeAssertion` methods. You can overwrite the existing implementations. See the following sample classes for examples:

SAML 1.x/WS-Federation

`AssertionSample.java`

SAML 2.0

`SAML2AssertionSample.java`

The sample classes are located in the directory `/sdk/samples/assertiongeneratorplugin`.

Note: The contents of the parameter string that your implementation passes into the `customizeAssertion` method is the responsibility of the custom object.

Deploy the Assertion Generator Plug-in

After you have coded your implementation class for the `AssertionGeneratorPlugin` interface, compile it and verify that SiteMinder can find your executable file.

To deploy the assertion generator plug-in

1. Compile the assertion plug-in Java file.

Compilation requires the following .jar files, which are installed with the Policy Server:

- `policy_server_home/bin/jars/SmJavaApi.jar`
- `policy_server_home/bin/thirdparty/xercesImpl.jar`
- `policy_server_home/bin/endorsed/xalan.jar`

2. In the `JVMOptions.txt` file, modify the `-Djava.class.path` value so it includes the classpath for the plug-in. This modification enables the plug-in to be loaded with the modified classpath. Locate the `JVMOptions.txt` file in the directory `installation_home\siteminder\config`.

Note: Do not modify the classpath for `xercesImpl.jar`, `xalan.jar`, or `SMJavaApi.jar`.

3. Enable the plug-in.

Enable the Assertion Generator Plug-in

After writing an assertion generator plug-in and compiling it, enable the plug-in by configuring settings in the FSS Administrative UI. The UI parameters let SiteMinder 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 FSS Administrative UI.
2. Navigate to the General settings.
3. In the Assertion Generator Plug-in section, complete the following fields:

Java Class Name

Specify a Java class name for an existing plug-in

Parameter

(Optional) Specify a string of parameters that is passed to the plug-in specified in the Java Class Name field.

Note: Instead of specifying the assertion plug-in class and its parameters through the Administrative UI, you can use the Policy Management API (C or Perl) to integrate the plug-in. For more information, see the *SiteMinder Programming Guide for C* or the *SiteMinder Programming Guide for Java*.

4. Restart the Policy Server.

Restarting the Policy Server ensures that the latest version of the assertion plug-in is picked up after being recompiled.

Creating Links to Consumer Resources for Single Sign-on

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

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

```
http://producer_site/affwebservices/public/intersitetransfer?SMASSERTIONREF=QUERY
&NAME=
affiliate_name&TARGET=http://consumer_site/target_url?query_parameter_name%
3Dquery_parameter_value%26query_parameter_name%3Dquery_parameter_value&S
MCONSUMERURL=
http://consumer_site/affwebservices/public/samlcc&AUTHREQUIREMENT=2
```

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

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

The variables in the intersite transfer URLs are as follows:

producer_site

Specifies the website where the user is authenticated.

affiliate_name

Indicates the name of an affiliate configured in an affiliate domain.

consumer_site

Indicates the site that the user wants to visit from the producer site.

target_url

Target page at the consumer site.

The intersite transfer URLs that the user selects must contain the query parameters listed in the table that follows.

Note: Query parameters for the SAML artifact profile must use HTTP-encoding.

Query Parameter	Meaning
SMASSERTIONREF (required)	For internal use. The value is always QUERY. Do not change this value.
NAME (required)	Name of an affiliate configured in an affiliate domain.
TARGET (required)	The target URL at the consumer site.
SMCONSUMERURL (required only for the artifact profile)	The URL at the consumer site processes the assertion and authenticates the user. For SAML 1.x artifact binding, if a value is specified for the Assertion Consumer URL, it takes precedence over the value of this query parameter.
AUTHREQUIREMENT=2 (required only for the artifact profile)	For internal use. The value is always 2. Do not change this value.

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

Example of an intersite transfer URL for the artifact profile:

```
http://www.smartway.com/affwebservices/public/intersitetransfer?SMASSERTION
REF=QUERY&NAME
=ahealthco&TARGET=http://www.ahealthco.com:85/smartway/index.jsp&SMCONS
UMERURL=
http://www.ahealthco.com:85/affwebservices/public/samlcc&AUTHREQUIREMENT
=2
```

Example of an intersite transfer URL for the POST profile:

```
http://www.smartway.com/affwebservices/public/intersitetransfer?SMASSERTION
REF
=QUERY&NAME=ahealthco&TARGET=http://www.ahealthco.com/index.html
```

Choosing Whether to Protect the Intersite Transfer URL

The web pages that include the intersite transfer URL links can be part of a realm configured for persistent sessions, which SiteMinder protects. When a user selects one of the links on a protected page, SiteMinder presents the user with an authentication challenge. After the user logs in, a persistent session can be established, which is required to store a SAML assertion.

If you do not to protect these pages, the Producer directs an affiliate user without a SiteMinder session to an authentication URL. This URL prompts the user to log in to receive a SiteMinder session. You define the Authentication URL when you configure an affiliate in the FSS Administrative UI.

Note: To set up persistent sessions, configure the session server. Set up a session server using the Policy Server Management Console.

Chapter 13: Configure SiteMinder as a SAML 1.x Consumer

This section contains the following topics:

- [SAML 1.x Authentication Scheme Prerequisites](#) (see page 255)
- [How To Configure SiteMinder as a SAML 1.x Consumer](#) (see page 256)
- [SAML 1.x Authentication Schemes](#) (see page 256)
- [Configure SAML 1.x Artifact Authentication](#) (see page 261)
- [Configure SAML 1.x POST Profile Authentication](#) (see page 263)
- [Customize Assertion Processing with the Message Consumer Plug-in](#) (see page 265)
- [Redirect Users After Failed SAML 1.x Authentication Attempts](#) (see page 269)
- [Supply SAML Attributes as HTTP Headers](#) (see page 270)
- [Enable Client Certificate Authentication for the Back Channel\(optional\)](#) (see page 275)
- [How To Protect a Resource with a SAML 1.x Authentication Scheme](#) (see page 277)

SAML 1.x Authentication Scheme Prerequisites

There are several prerequisites you must fulfill before configuring a SiteMinder relying partner.

- Install the Policy Server.
For installation instructions, refer to the Policy Server Installation Guide.
- Install one of the following
 - The Web Agent and the Web Agent Option Pack. The Web Agent authenticates users and establishes a SiteMinder session. The Option Pack provides the Federation Web Services application. Be sure to deploy the FWS application on the appropriate system in your network.
 - The SPS federation gateway, which has an embedded Web Agent and has the Federation Web Services application on the embedded Tomcat web server.

For more information, see the *Web Agent Option Pack Guide* or the *Secure Proxy Server Administration Guide*.

- Set up a key database for each Policy Server that is responsible for signing, verification or both. Import private keys and certificates for functions that require verification and encrypting of messages.

The key database is a flat-file key and certificate database that lets you manage and retrieve keys and certificates required to sign and validate SAML responses used with SAML POST profile authentication.
- An asserting partner is set up within the federated network.

How To Configure SiteMinder as a SAML 1.x Consumer

Configuring SiteMinder as SAML 1.x consumer requires the following tasks:

1. Complete the SAML 1.x authentication scheme prerequisites.
2. Select the authentication scheme type and assign it a name.
3. Specify the namespace for users being authenticated with the SAML 1.x authentication scheme.
4. Select the single sign-on profile that this consumer supports (artifact or POST).
5. Configure a SAML authentication scheme for each Producer that is a federation partner and generates assertions. Bind each scheme to a realm. The realm must contain the target URLs for federated resources. Protect these resources with a SiteMinder policy.

Tips:

- Certain parameter values at the Producer and Consumer must match for the configuration to work. A list of those parameters is available in [Configuration Settings that Must Use the Same Values](#) (see page 471).
- Verify that you are using the correct URLs for the Federation Web Services servlets. The URLs are listed in [Federation Web Services URLs Used in SiteMinder Configuration](#) (see page 477).

Optional Tasks to Configure a SiteMinder Consumer

The following tasks are optional for configuring SiteMinder as a consumer:

- Customize assertions using the message consumer plug-in.
- Redirect failed authentication attempts.

SAML 1.x Authentication Schemes

A consumer is a site that uses a SAML 1.x assertion to authenticate a user.

Note: A site can be a SAML producer and a SAML consumer.

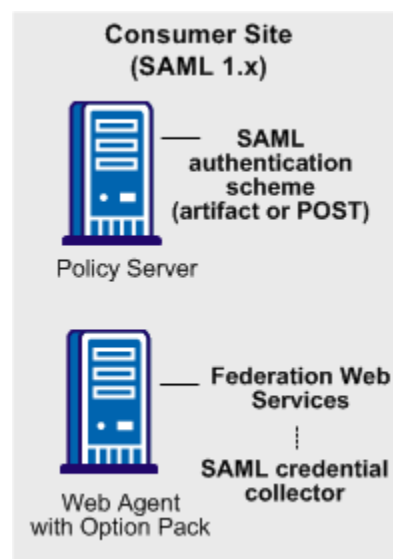
Any SiteMinder site with Federation Security Services functionality can consume SAML 1.x assertions and can use these assertions to authenticate users. When an assertion is consumed, the site has to be able to compare the information from the assertion against a user directory to complete the authentication process.

SiteMinder provides the following SAML 1.x authentication methods:

- SAML Artifact profile
- SAML POST profile

The SAML-based authentication schemes let a consumer site authenticate a user. Consuming a SAML assertion and establishing a SiteMinder session enables cross-domain single sign-on. After the user is identified, the consumer site can authorize the user for specific resources.

The following illustration shows the major components for authentication at the consumer site.



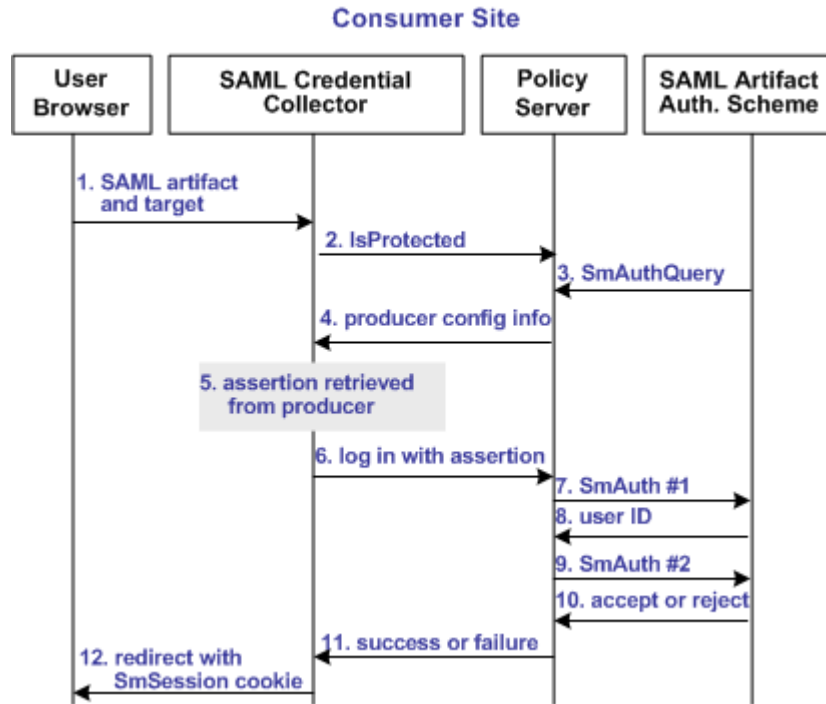
Note: The SPS federation gateway can replace the Web Agent and Web Agent Option Pack to provide the Federation Web Services application functions. For information about installing and configuring the SPS federation gateway, see the *Secure Proxy Server Administration Guide*.

The SAML 1.x authentication scheme is configured at the consumer-side Policy Server. The SAML credential collector is a component of the Federation Web Services application. The credential collector is installed on the consumer-side Web Agent, or on an SPS federation gateway. The credential collector obtains information from the SAML authentication scheme at the Policy Server, then uses that information to access a SAML assertion.

The SAML assertion becomes the credentials that grant access to the Policy Server at the consumer site. The user is authenticated and authorized, and if authorization is successful, the user is redirected to the target resource.

SAML 1.x Artifact Authentication Scheme Overview

The following illustration shows how the SAML 1.x artifact authentication scheme processes requests.



Note: An SPS federation gateway, or the Web Agent and Web Agent Option Pack, provide the Agent and SAML Credential Collector functionality.

Unless otherwise stated, all activity in this process occurs at the Consumer site:

1. A user is redirected to the SAML credential collector with a SAML artifact and a target URL.
The artifact and target URL are originally generated from the Web Agent at the producer site.
2. The SAML credential collector calls the Policy Server to determine whether the SAML artifact authentication scheme protects the requested resource.
3. The Policy Server passes the necessary data to the SAML artifact authentication scheme, which extracts the producer configuration information.
4. The Policy Server returns the producer configuration information to the SAML credential collector. This information enables the credential collector servlet to call a producer site and retrieve a SAML assertion.
5. The SAML credential collector takes the data from the Policy Server and uses it to retrieve the SAML assertion.

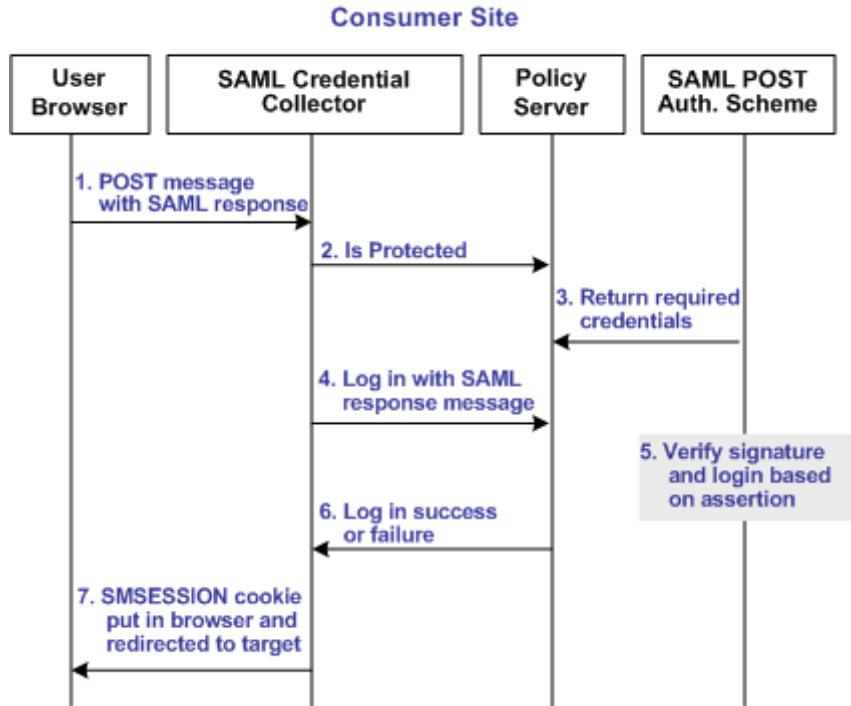
6. Once an assertion is returned, the credential collector uses the assertion as credentials, and logs in to the Policy Server.
7. The Policy Server makes the initial user disambiguation call to the SAML authentication scheme.
8. Using the authentication scheme data and the assertion, the scheme locates the user and returns a unique identifier for the user to the credential collector.
9. The Policy Server makes the second user authentication call to the authentication scheme.

Note: The SiteMinder Authentication AP dictate the two-step authentication process. For more information, see the *SiteMinder Programming Guide for C* or the *SiteMinder Programming Guide for Java*.

10. The scheme validates the SAML assertion and returns an accept or reject message to the Policy Server.
11. The Policy Server sends the accept or reject message to the credential collector.
12. The SAML credential collector creates a session cookie and places it in the browser, and then redirects the user to the target resource. If the login fails, the credential collector redirects the user to a No Access URL.

SAML 1.x POST Profile Authentication Scheme Overview

The following illustration shows how the SAML 1.x POST profile authentication scheme processes requests.



Note: The SPS federation gateway or the Web Agent Option Pack provide the SAML Credential Collector functionality.

Unless otherwise stated, the following process takes place at the consumer site:

1. A browser posts an HTML form to the SAML credential collector URL. This form contains a SAML response message and the address of the target URL, originally generated at the producer.
2. The SAML credential collector contacts the Policy Server to determine whether the target resource is protected.
3. The Policy Server replies that the SAML POST profile authentication scheme protects the target URL. A signed response from the posted form is the expected credential for the login call.
4. The SAML credential collector makes a login call to the Policy Server, passing the digitally signed SAML response as credentials.
5. The SAML POST profile authentication scheme verifies the signature and other fields of the response and the assertion.

6. If the checks succeed and the user is found in the directory, then authentication succeeds. If any of the checks fail, authentication fails.
7. The SAML credential collector creates an SMSESSION cookie. This cookie is put in the browser and the user is redirected to the target resource. If the login fails, the credential collector redirects the user to the configured No Access URL.

Configure SAML 1.x Artifact Authentication

Before you can assign a SAML artifact authentication scheme to a realm, you must configure the scheme.

To configure the SAML artifact authentication scheme:

1. Check the [SAML 1.x Authentication Scheme Prerequisites](#) (see page 255).
2. Log into the FSS Administrative UI.
3. From the menu bar, select Edit, System Configuration, Create Authentication Scheme.

The Authentication Scheme Properties dialog box opens.

4. Fill in the fields for the:
 - Scheme Common Setup group box
 - Scheme Setup tab
 - The Advanced tab (optional)

Configure the SAML 1.x Artifact Scheme Setup

The configuration of the SAML 1.x artifact authentication scheme lets you enter information about the producer site that provides the SAML assertion to the consumer.

After you configure an authentication scheme, associate the scheme with a realm that contains the resource you want to protect.

To configure the SAML 1.x artifact authentication scheme

1. From the Authentication Scheme Type drop-down list, select SAML Artifact Template.

The contents of the SiteMinder Authentication Scheme dialog change to support the SAML artifact scheme.

2. Configure the scheme setup.

Note: Click Help for descriptions of settings and controls, including their respective requirements and limits.

Important! The Affiliate Name, Password, and Verify Password fields must match other values in your federation network. For details, go to [Configuration Settings that Must Use the Same Values](#) (see page 471).

3. (Optional) Select Additional Configuration to configure features such as the Message Consumer API, redirect URLs for authentication errors, and to specify the target resource at the consumer.

Note: You can specify the target resource using the value of the TARGET query parameter in the authentication response URL or by specifying a default target URL in this dialog. The checkbox labeled Query Parameter TARGET Overrides Default Target URL is selected by default. If you uncheck this box, you must enter a value for the Default TARGET URL field.

4. Click OK to save the scheme.

The SAML 1.x Artifact authentication scheme is now configured.

For the SAML artifact profile, the producer sends the assertion to the consumer over a protected backchannel. If you are using basic authentication to protect the backchannel, the value of the Affiliate Name field is the name of the consumer. If you are using client certificate authentication for the backchannel, the value of the Affiliate Name field must be the alias of the client certificate stored in the smkeydatabase.

If you use client certificate authentication for communication over the backchannel, you can use non-FIPS 140 encrypted certificates even if the Policy Server is operating in FIPS-only mode. However, for a strictly FIPS-only installation, use certificates only encrypted with FIPS 140-compatible algorithms.

More Information:

[How To Protect a Resource with a SAML 1.x Authentication Scheme](#) (see page 277)
[Modify the Key Database Using smkeytool](#) (see page 195)

Create a Custom SAML Artifact Authentication Scheme (Optional)

The Advanced tab of the Authentication Scheme dialog box lets you use a custom SAML artifact scheme written with the SiteMinder Authentication API.

Complete the following fields:

- Library
- Parameter

Backchannel Configuration for HTTP-Artifact SSO

For the SAML artifact profile, the asserting party sends the assertion to the consumer over a back channel. Protect the back channel with an authentication scheme. You can use a basic or client certificate authentication scheme to secure the back channel.

- Basic authentication

If you use basic authentication and SiteMinder is at both partners, the Affiliate Name at each site is the name of the consumer. If the asserting party is not SiteMinder, the asserting party administrator must provide you with the name they are using to identify your site. Specify the supplied name as the Affiliate Name in your authentication scheme configuration.

- Client certificate authentication

If you use client certificate authentication for the back channel, the affiliate name in the Administrative UI must be the alias of the client certificate. Additionally, the CN of the certificate subject must also match the affiliate name. Matching the affiliate name, alias and CN is required.

The Policy Server supports client certificate authentication over the backchannel using non-FIPS 140 encrypted certificates, even when the Policy Server is operating in FIPS-only mode. However, for a strictly FIPS-only installation, use certificates only encrypted with FIPS 140-compatible algorithms.

The client certificate is stored in the certificate data store.

Configure SAML 1.x POST Profile Authentication

To configure the SAML POST profile authentication scheme

1. Check the [SAML 1.x Authentication Scheme Prerequisites](#) (see page 255).
2. Log into the FSS Administrative UI.

3. From the menu bar, select Edit, System Configuration, Create Authentication Scheme.

The Authentication Scheme Properties dialog box opens.

4. Fill in the fields for the:
 - Scheme Common Setup group box
 - Scheme Setup tab
 - The Advanced tab (optional)

More Information:

[Create the SAML 1.x POST Common Setup and Scheme Setup](#) (see page 264)
[Configure a Custom SAML 1.x POST Authentication Scheme](#) (see page 265)

Create the SAML 1.x POST Common Setup and Scheme Setup

Before you can assign a SAML POST profile authentication scheme to a realm, you must configure the scheme. The Scheme Setup tab is where you enter information about the producer site that provides the SAML assertion to the consumer.

After configuring an authentication scheme, associate the scheme with a realm that contains the resource you want to protect.

To configure the SAML 1.x POST authentication scheme

1. From the Authentication Scheme Type drop-down list, select SAML POST Template.
The contents of the SiteMinder Authentication Scheme dialog box change to support the SAML POST profile scheme
2. Configure the scheme setup by configuring the fields on the tab.

Important! The Affiliate Name, Password, and Verify Password fields must match other values in your federation network. For details, go to [Configuration Settings that Must Use the Same Values](#) (see page 471).

Note: Click Help for descriptions of settings and controls, including their respective requirements and limits.

3. (Optional) Select Additional Configuration to configure features such as the Message Consumer API, redirect URLs for authentication errors, and to specify the target resource at the consumer.

Note: You can specify the target resource using the value of the TARGET query parameter in the authentication response URL or by specifying a default target URL in this dialog. The checkbox labeled Query Parameter TARGET Overrides Default Target URL is selected by default. If you uncheck this box, you must enter a value for the Default TARGET URL field.

4. Click OK to save the scheme.

The SAML 1.x POST authentication scheme is now configured.

More Information:

[How To Protect a Resource with a SAML 1.x Authentication Scheme](#) (see page 277)

Configure a Custom SAML 1.x POST Authentication Scheme

The Advanced tab of the Authentication Scheme dialog box lets you use a custom SAML POST scheme written with the SiteMinder Authentication API.

Complete the following fields:

- Library
- Parameter

Customize Assertion Processing with the Message Consumer Plug-in

The message consumer plug-in is a Java program that implements the Message Consumer Plug-in. The plug-in lets you implement your own business logic for processing assertions, such as rejecting an assertion and returning a status code. This additional processing works together with the standard processing of an assertion.

Note: For more information about status codes for authentication and disambiguation, see the *SiteMinder Programming Guide for Java*.

During authentication, SiteMinder first tries to process the assertion by mapping a user to its local user store. If SiteMinder cannot find the user, it calls the `postDisambiguateUser` method of the message consumer plug-in.

If the plug-in successfully finds the user, SiteMinder proceeds to the second phase of authentication. If the plug-in cannot map the user to a local user store, the plug-in returns a `UserNotFound` error. The plug-in can optionally use the redirect URL feature. Without the consumer plug-in, the redirect URLs are based on the error that the SAML authentication scheme generates.

During the second phase of authentication, SiteMinder calls the `postAuthenticateUser` method of the message consumer plug-in, if the plug-in is configured. If the method succeeds, SiteMinder redirects the user to the requested resource. If the method fails, you can configure the plug-in to send the user to a failure page. The failure page can be one of the redirect URLs that you can specify with the authentication scheme configuration.

Additional information about the message consumer plug-in can be found as follows:

- Reference information (method signatures, parameters, return values, data types), and the constructor for `UserContext` class, are in the *Java Developer Reference*. Refer to the `MessageConsumerPlugin` interface.
- Overview and conceptual information about authentication and authorization APIs, see the *SiteMinder Programming Guide for Java*.

To configure the plugin

1. Install the SiteMinder SDK, if you have not done so already.
2. Implement the `MessageConsumerPlugin.java` interface, which is part of the SiteMinder SDK.
3. Deploy your message consumer plug-in implementation class.
4. Enable the message consumer plug-in in the Administrative UI.

Implement the `MessageConsumerPlugin` Interface

Create a custom message consumer plug-in by implementing the `MessageConsumerPlugin.java` interface. The minimum requirements for the implementation class are listed in the following procedure.

Follow these steps:

1. Provide a public default constructor method that contains no parameters.
2. Provide code so that the implementation is stateless. Many threads must be able to use a single plug-in class.

3. Implement methods in the interface as your requirements demand.

The MessageConsumerPlugin includes the following four methods:

init()

Performs any initialization procedures that the plug-in requires. 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. SiteMinder calls this method once for each plug-in instance, when SiteMinder is shutting down.

postDisambiguateUser()

Provides processing to disambiguate a user when the authentication scheme is unable to do so. Alternatively, this method can add data for new federation users to a user store. This method receives the decrypted assertion. The decrypted assertion is added to the properties map passed to plug-in under the key "_DecryptedAssertion".

postAuthenticateUser()

Provides additional code to determine the final outcome of assertion processing, regardless of whether the Policy Server processing is a success or failure.

SiteMinder provides the following samples of the Message Consumer plug-in class:

MessageConsumerPluginSample.java in
installation_home\sdk\samples\messageconsumerplugin

MessageConsumerSAML20.java in
installation_home\sdk\samples\authextensionsaml20

Deploy a Message Consumer Plug-in

After you have coded your implementation class for the MessageConsumerPlugin interface, compile it and verify that SiteMinder can find your executable file.

To deploy the Message Consumer Plugin:

1. Compile the MessageConsumerPlugin Java file. The file requires the following dependent libraries, which are installed with the Policy Server:

installation_home\siteminder\bin\jars\SmJavaApi.jar

An identical copy of SmJavaApi.jar is installed with SiteMinder SDK. The file is in the directory *installation_home*\sdk\java\SmJavaApi.jar.

You can use either of them at development time.

2. When a plug-in class is available, in a folder or a jar file, modify the `-Djava.class.path` value in the `JVMOptions.txt` file. This step enables the plug-in class to load with the modified classpath. Locate the `JVMOptions.txt` file in the directory `installation_home\siteminder\config`.

Note: Do not modify the classpath for the existing `xerces.jar`, `xalan.jar`, or `SmJavaApi.jar`.

3. Restart the Policy Server to pick up the latest version of `MessageConsumerPlugin`. This step is necessary each time the plug-in Java file is recompiled.
4. Enable the plug-in.

Enable the Message Consumer Plug-in for SAML 1.x

After writing a message consumer plug-in, enable the plug-in by configuring settings in the FSS Administrative UI.

Note: Specify a Message Consumer plug-in for each authentication scheme.

Do not configure the plug-in settings until you deploy the plug-in.

Follow these steps:

1. Log on to the FSS Administrative UI.
2. Navigate to the SAML Auth Scheme Properties dialog and click Additional Configuration dialog.
3. Complete the following fields:

Full Java Class Name

Specify the Java class name for the plug-in, For example, a sample class included with the SiteMinder SDK is:

`com.ca.messageconsumerplugin.MessageConsumerPluginSample`

Parameter

Specify a string of parameters to be passed to the plug-in specified in the Full Java Class Name field.

4. Click OK to save the changes.
You return to the Authentication Scheme Properties dialog.
5. Click OK to exit the dialog.
6. Restart the Policy Server.

As an alternative to configuring the plug-in in the UI, use the Policy Management API (C or Perl) to set the `IdpPluginClass` and `IdpPluginParameters`.

Additional information about the message consumer plug-in can be found as follows:

- Reference information (method signatures, parameters, return values, data types) are in the *Java Developer's Reference*. Refer to the MessageConsumerPlugin interface.
- Overview and conceptual information about authentication and authorization APIs, see the *SiteMinder Programming Guide for Java*.

Redirect Users After Failed SAML 1.x Authentication Attempts

For single sign-on processing, you can configure several optional redirect URLs if authentication at the consumer fails. The redirect URLs allow finer control over where a user is redirected if the assertion is not valid. For example, if a user cannot be located in a user store, you can fill in a User Not Found redirect URL and send the user to a registration page.

Note: These URLs are not required.

If you do not configure redirect URLs, standard SiteMinder processing takes place. How a failed authentication is handled depends on the configuration of the authentication scheme.

To configure optional redirect URLs

1. From the Authentication Scheme Properties dialog, click Additional Configuration. The SAML 1.x Auth Scheme Properties dialog opens.
2. Fill in a URL for one or more of the following fields:
 - Redirect URL for the User Not Found status
 - Redirect URL for the invalid SSO Message status
 - Redirect URL for the Unaccepted User Credential (SSO Message) status

If you enter a value for the Redirect URL, you must also choose a mode.

Note: Click Help for descriptions of settings and controls, including their respective requirements and limits.

Federation Web Services handles the errors by mapping the authentication reason into one of the configured redirect URLs, then the user can be redirected to that URL to report the error.

Note: These redirect URLs can be used in conjunction with the SiteMinder Message Consumer Plug-in for further assertion processing. If authentication fails, the plug-in can send the user to one of the redirect URLs you specify.

Supply SAML Attributes as HTTP Headers

An assertion response can include attributes in the assertion. These attributes can be supplied as HTTP header variables so a client application can use them for finer grained access control.

The benefits of including attributes in HTTP headers are as follows:

- HTTP headers are not persistent. They are present only within the request or response that contains them.
- HTTP headers, as supplied by the SiteMinder Web Agent, are not visible in the browser, which reduces security concerns.

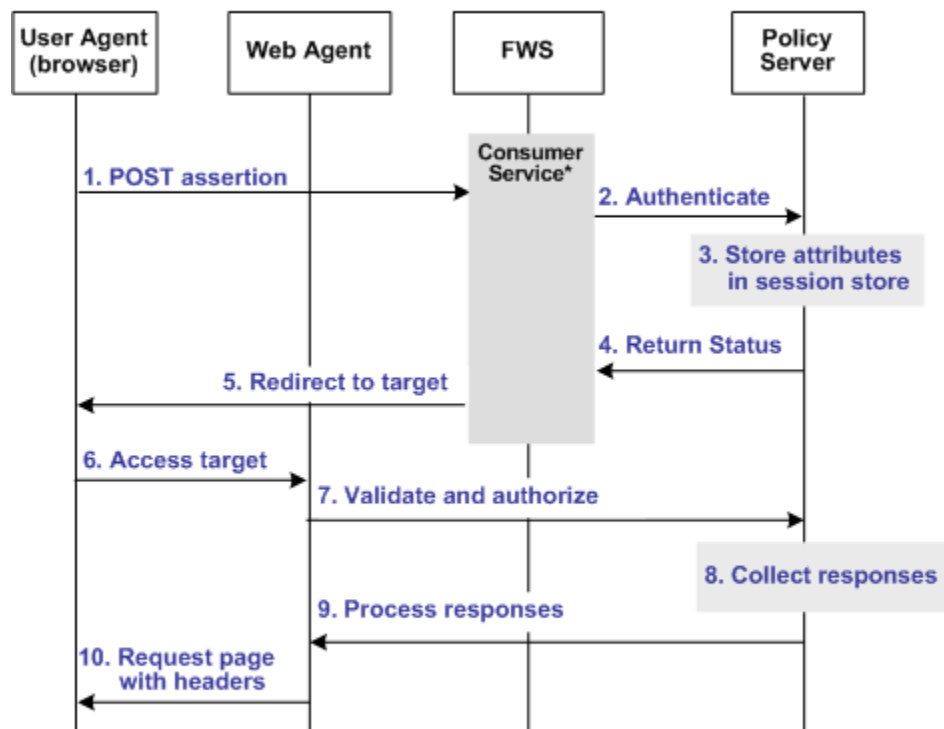
Note: The HTTP headers have size restrictions that the attributes cannot exceed. SiteMinder can send an attribute in a header up to the web server size limit for a header. Only one assertion attribute per header is allowed. See the documentation for your web server to determine the header size limit.

Use Case for SAML Attributes As HTTP Headers

During authentication, a series of SAML attributes are extracted from an assertion and supplied as HTTP headers. During the authorization process, these headers are returned to the customer application.

The following flow diagram shows the sequence of events at runtime:

Processing Headers as Attributes at the Consumer



*Consumer service can be one of the following:
 –SAML Credential Collector (SAML 1.x)
 –Assertion Consumer Service (SAML 2.0)
 –Security Token Consumer Service (WS-Federation)

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

To process the attributes as HTTP headers, the sequence of events is as follows:

1. After the assertion is generated at the asserting party, it sends the assertion to the appropriate consumer service at the relying party. The delivery mechanism (POST or Artifact or WS-Fed) is irrelevant.

Note: The consumer service can be the SAML credential collector (SAML 1.x), the Assertion Consumer Service (SAML 2.0), or Security Token Consumer Service (WS-Federation).

2. The consumer service calls its local Policy Server to use the configured authentication scheme to authenticate the user with the assertion.

3. If the authentication scheme redirect mode parameter is set to PersistAttributes, the Policy Server caches the attributes in the session store as session variables.
4. The result of the authentication is returned to the consumer service.
5. The consumer service redirects the browser to the protected target resource.
6. The browser tries to access the target resource.
7. The Web Agent calls the Policy Server to validate the user session and to verify that the user is authorized to access the target resource.
8. The Policy Server retrieves the attributes by a configured response.
9. The Policy Server processes the responses and sends the attributes to the Web Agent.
10. The Web Agent sets the HTTP headers as necessary.

Configuration Overview to Supply Attributes as HTTP Headers

Several configuration steps are required to retrieve the SAML attributes cached in the session store and provide them as HTTP headers.

Follow these steps:

1. Select PersistAttributes as the redirect mode for the SAML authentication scheme, which enables the SAML Attributes to be returned as HTTP headers.
2. Configure an authorization rule for the realm that contains the target resource.
3. Set PersistentRealm in the realm protecting the target resource.
4. Configure a response that uses the active response type for each SAML attribute to be supplied as a header.
5. Create a policy that binds the authorization rule and active response to implement the user of attributes as HTTP headers.

Set the Redirect Mode to Store SAML Attributes

After the consuming partner authenticates the user with the SAML assertion, the SAML attributes are written to the session store. The browser is then redirected to the target resource.

To redirect the browser with the attribute data

1. Log in to the FSS Administrative UI.
2. Access the SAML authentication scheme properties dialog.
The properties dialog opens.

3. Set the Redirect Mode parameter to PersistAttributes.

For SAML 1.x, the Redirect Mode is on the Scheme Setup tab. For SAML 2.0 and WS-Federation, the Redirect Mode is on the SSO tab accessed from the authentication scheme properties dialog.

4. Click OK to save your changes.

The redirect mode is now set to pass on the attribute data.

Create an Authorization Rule to Validate Users

For the realm containing the protected target resource, create a rule that is triggered during the authorization process to retrieve the SAML attributes from the session store.

The rule is based on an authorization event (`onAccessAccept`). Because the user has already been authenticated by the FWS application, the Web Agent cannot reauthenticate the user and pass on the HTTP headers. The retrieval of the attributes happen during the authorization stage.

To create an `OnAccessAccept` Rule for the realm

1. Log on to the FSS Administrative UI.
2. From the Domains tab, navigate to the realm which protects the target resource.
3. Select the realm with the target resource and select Create Rule under Realm.
The Rule Properties dialog opens.
4. Enter a name in the Name field that describes the rules purpose as an authorization rule.
5. Select the realm protecting the target resource for the Realm field.
6. Enter an asterisk (*) in the Resource field.
7. Select Authorization events and `OnAccessAccept` in the Action section.
8. Verify that Enabled is selected in the Allow/Deny and Enable/Disable section.
9. Click OK to save the rule.

The authorization rule is now defined for the realm with the protected resource.

Configure a Response to Send Attributes as HTTP Headers

Configure a response that sends the SAML attributes as HTTP headers to the Web Agent. The Web Agent processes the response and makes the header variables available to the client application.

To create a response to send the attributes as headers

1. Log on to the FSS Administrative UI.
2. From the Domains tab, select the domain that contains the target resource.
3. In the domain, select the Responses object and create a response.
The Response Properties dialog opens.
4. Click Create.
The Response Attribute dialog opens.
5. Select WebAgent-HTTP-Header-Variable in the Attribute field.
6. Select Active Response in the Attribute Kind section.
7. Complete the fields in the Attribute Fields section as follows:

Variable Name

Specify the name you want for the header variable. You assign this name.

Library Name

smfedattrresponse

This value must be the entry for this field.

Function Name

getAttributeValue

This value must be the entry for this field.

Parameters

Specify the name of the attribute as it appears in the assertion.

An agreement between you and your federated partner determines the attributes that are in the assertion.

8. Click on OK to save the attribute.
9. Repeat the procedure for each attribute that must become an HTTP header variable. You can configure many attributes for a single response.

The response sends the attributes on to the Web Agent to become HTTP headers.

Create a Policy to Implement Attributes as HTTP Headers

To implement the use of SAML attributes as HTTP headers, you must group together the authorization event rule and active response in a policy.

To create the policy to generate HTTP Headers from SAML attributes

1. Log on to the FSS Administrative UI.
2. From the Domains tab, select the domain that contains the target resource.
3. In the domain, select the Policies object and create a policy.
The Policy Properties dialog opens.
4. Enter a descriptive name in the Name field.
5. Select the users that must have access to the protected resource in the Users tab.
6. Add the authorization rule you created previously on the Rules tab.
7. Select the authorization rule and click Set Response.
The Available Responses dialog opens.
8. Select the active response you created previously and click OK.
You return to the Rules tab. The response appears with the authentication rule.
9. Click OK to save the policy.

The policy that enables SAML attributes to be used as HTTP headers is complete.

Enable Client Certificate Authentication for the Back Channel(optional)

If you have configured single sign-on with the artifact profile, you can select client certificate authentication to protect the Assertion Retrieval Service at the producer. This service retrieves the assertion and sends it to the consumer.

Note: Client certificate authentication is optional; you can also use Basic authentication.

The SAML credential collector invokes the SAML artifact authentication scheme. The SAML credential collector collects information from the scheme to retrieve the SAML assertion from the Producer. You are required to specify the authentication method for the realm that contains the Assertion Retrieval Service. The SAML credential collector determines what type of credentials to provide to retrieve the assertion.

If the Assertion Retrieval Service is part of a realm using a client certificate authentication scheme, complete these configuration tasks:

- At the Consumer, select the client certificate option to indicate that a certificate provides credentials.
- At the Producer, create a policy to protect the Assertion Retrieval Service.

The process of enabling client certificate authentication includes the following:

1. Add a client certificate to the certificate data store.
2. Select the client certificate option for back channel authentication.

Add a Client Certificate to smkeydatabase

When you are adding a client certificate to the key database, note the following:

- The value of `dname`, which specifies the Consumer name, can be any attribute from the Consumer subject DN. The Policy Server at the producer site can use its certificate mapping functionality to map the Consumer to a local user directory entry.
- The value for `alias` associated with the private key is the same as the value of the Affiliate Name field. The Attribute Name field is in the Scheme Setup section of SAML Artifact Authentication scheme dialog. The attribute of the Consumer subject DN, represented in the example by the CN value also reflects the Affiliate Name value.

For example, if you entered CompanyA as the Affiliate Name, then `alias` is Company A. The attribute is `CN=CompanyA, OU=Development, O=CA, L=Waltham, ST=MA, C=US`

- To refer to the existing key store entry, subsequent `keytool` commands use the same `alias`.
- The value for `password` is same as the value of the Password field specified in the Scheme Setup dialog for the SAML Artifact Authentication Scheme.

To create and store a client certificate in the smkeydatabase file at the Consumer

1. Open a command window.
2. If necessary, create a key database by entering:

```
smkeytool -createDB -password fedDB
```

3. Generate a key-pair combination.

For example, to create a private key using the PKCS8 format enter:

```
smkeytool -addPrivKey -alias CompanyA -keyfile idp1pkey.pkcs8 -certfile idp1.crt  
-password smdb
```

This example assumes that you are running smkeytool from the directory where the certificate and key are located, so there are no file paths necessary.

The certificate is now added to the smkeydatabase.

4. Restart the Policy Server to see the smkeydatabase changes immediately.

Select the Client Cert Option for Authentication

For the consumer to present a certificate as credentials when trying to access the Assertion Retrieval Service at the producer, select the client certificate option.

To select the client certificate option:

1. Go to the Scheme Setup tab of the SAML Artifact Authentication scheme dialog box.
2. Select Client Cert for the Authentication field.

How To Protect a Resource with a SAML 1.x Authentication Scheme

Protect target federation resources by configuring a SiteMinder policy that uses the SAML 1.x authentication scheme.

Follow these steps:

1. Create a realm that uses the SAML authentication scheme. The realm is the collection of target resources.

You can create a realm in the following ways:

- Create a unique realm for each authentication scheme already configured.
- [Configure a single target realm](#) (see page 279) that uses a custom authentication scheme to dispatch requests to the corresponding SAML authentication schemes. Configuring one realm with a single target for all producers simplifies configuration of realms for SAML authentication.

2. Configure an associated rule and optionally, a response.
3. Group the realm, rule, and response into a policy that protects the target resource.

Important! Each target URL in the realm is also identified in an intersite transfer URL. The intersite transfer URL redirects a user from the producer to the consumer. You specify this URL in the URL TARGET variable. At the producer site, an administrator includes this URL in a link that redirects the user to the consumer.

Configure a Unique Realm for Each SAML Authentication Scheme

The procedure for configuring a unique realm for each SAML authentication scheme (artifact or profile) follows the standard instructions for creating realms in the FSS Administrative UI.

To create a realm for each SAML authentication scheme

1. Log on to the FSS Administrative UI.
2. Click the System tab.
3. Click Edit, System Configuration, Create Domain.
The Domain dialog opens.
4. Create a policy domain that will contain the realm with the target resources.
5. Create a realm under the policy domain you created in the previous step, noting the following:
 - a. Select the Web Agent protecting the web server where the target federation resources reside for the Agent field.
 - b. Select the SAML authentication scheme for the Authentication Scheme field. This is the SAML scheme that should protect the realm.
6. Create a rule for the realm.

As part of the rule you select a Web Agent action (Get, Post, or Put), which allows you to control processing when users authenticate to gain access to a resource.

Note: Click Help for descriptions of settings and controls, including their respective requirements and limits.

The realm is now configured.

Form the Policy to Protect the Target Resource

After you create the realm, you add it to a policy that protects target federation resources.

Note: The following procedure assumes that a user directory has already been created.

To create a policy for the target federation resources

1. Log on to the FSS Administrative UI.
2. Expand the domain with the target realm.
3. Select the Policies object.
The Policy Properties dialog opens.
4. Configure the policy, using the realm you previously created for federation resources.
5. Save the policy.
6. Exit the FSS Administrative UI.

For more information about creating policies, see the *Policy Server Configuration Guide*.

Configure a Single Target Realm for All Authentication Schemes

To simplify configuration of realms for authentication schemes, create a single target realm for multiple sites generating assertions.

To do this task, set up the following components:

- A single custom authentication scheme
This custom scheme forwards requests to the corresponding SAML or WS-Federation authentication schemes that you already configured for each asserting party.
- A single realm with one target URL

More information:

[Create the Custom Authentication Scheme](#) (see page 280)

[Configure the Single Target Realm](#) (see page 282)

Create SAML Authentication Schemes for the Single Target Realm

Configure the necessary SAML authentication schemes that will be referenced by the custom authentication scheme associated with the single target realm. When you define the custom authentication scheme, you define a parameter that instructs the Policy Server which SAML authentication schemes the custom scheme can apply to resource requests.

To create the SAML authentication scheme

1. Log on to the FSS Administrative UI.
2. Create SAML authentication schemes according to the procedures in this guide for the SAML protocol you are using.
3. Exit the FSS Administrative UI.

More information:

[SAML 1.x Authentication Schemes](#) (see page 256)

[SiteMinder as a Service Provider](#) (see page 347)

Create the Custom Authentication Scheme

A single target realm relies on a specific custom authentication scheme to work properly.

To configure a custom authentication scheme for a single target realm

1. Log on to the FSS Administrative UI.
2. Select the System tab.
3. Select Edit, System Configuration, Create Authentication Scheme.

The Authentication Scheme Properties dialog opens.

4. Complete the fields as follows:

Note: Click Help for descriptions of settings and controls, including their respective requirements and limits.

Name

Enter a descriptive name to indicate this is a custom auth scheme, such as SAML Custom Auth Scheme.

5. Complete the following field in the Scheme Common Setup section:

Authentication Scheme Type

Custom Template

6. Complete the following fields in the Scheme Setup tab

Library

smauthsinglefed

Secret and Confirm Secret

Leave this field blank.

Confirm Secret

Leave this field blank

Parameter

Specify one of the following:

- SCHEMESET=LIST; <saml-scheme1>;<saml_scheme2>

Specifies the list of SAML authentication scheme names to use. If you configured an artifact scheme called artifact_producer1 and POST profile scheme called samlpost_producer2, you will enter these schemes. For example:

SCHEMESET=LIST;artifact_producer1;samlpost_producer2

- SCHEMESET=SAML_ALL;

Specifies all the schemes you have configured. The custom authentication scheme will enumerate all the SAML authentication schemes and find the one with the correct Provider Source ID for the request.

- SCHEMESET=SAML_POST;

Specifies all the SAML POST Profile schemes you have configured. The custom authentication scheme will enumerate the POST Profile schemes and find the one with the correct Provider Source ID for the request.

- SCHEMESET=SAML_ART;

Specifies all the SAML artifact schemes you have configured. The custom authentication scheme will enumerate the artifact schemes and find the one with the correct Provider Source ID for the request.

Enable this scheme for SiteMinder Administrators

Leave unchecked.

7. Click OK to save your changes.

Configure the Single Target Realm

After you configure the authentication schemes, including the custom authentication scheme, you can configure a single target realm for federation resources.

To create the single target realm

1. Log in to the FSS Administrative UI.
2. Select the Domains tab.
3. Select the policy domain you previously created for the single target realm.
4. Select the Realms object and select Edit, Create Realm.

The Realm Properties dialog opens.

5. Enter the following values to create the single target realm:

Note: Click Help for descriptions of settings and controls, including their respective requirements and limits.

Name

Enter a name for this single target realm.

6. Complete the following field in the Resource section:

Agent

Select the SiteMinder Web Agent protecting the web server with the target resources.

Resource Filter

Specify the location of the target resources. Any user requesting a federated resource is be redirected to this location.

For example, /FederatedResources.

7. Select the Protected option in the Default Resource Protection section.
8. Select the previously configured custom authentication scheme in the Authentication Scheme section. This custom authentication uses the smauthsinglefed library.

For example, if the custom scheme was named Fed Custom Auth Scheme, you must select this scheme.
9. Click OK.

The single target realm task is complete.

Configure the Rule for the Single Target Realm

Under the single target realm, configure a rule to protect the resources.

1. Select the single target realm.
2. Select Edit, *single target realm*, Create Rule under Realm.

The Rule Properties dialog displays.

3. Enter values for the fields in the dialog.

Note: Click Help for descriptions of settings and controls, including their respective requirements and limits.

4. Click OK.

The rule for the single target realm configuration is created. It can now be used in a policy.

Create a Policy Using the Single Target Realm

Create a policy that references the single target realm. Remember that the single target realm uses the custom authentication scheme that directs requests to the appropriate SAML/WS-Fed authentication scheme.

Note: This procedure assumes that you have already configured the domain, custom authentication scheme, single target realm and associated rule.

To create a policy and add it to an existing domain

1. Log on to the FSS Administrative UI.
2. From the Domains tab, select the Policies object.
3. Click Edit, Create Policy.
The Policy Properties dialog opens.
4. Enter a name and a description of the policy in the General section.
5. Add users to the policy from the Users tab.
6. Add the rule you created for the single target realm from the Rules tab.
The remaining tabs are optional.
7. Click OK.

The policy task is complete.

Chapter 14: Configure SiteMinder as a SAML 2.0 Identity Provider

This section contains the following topics:

- [Prerequisites for a SiteMinder Asserting Party](#) (see page 285)
- [Configuration Checklist at the Identity Provider](#) (see page 286)
- [How to Configure a SiteMinder Identity Provider](#) (see page 286)
- [Add a SAML 2.0 Service Provider to an Affiliate Domain](#) (see page 287)
- [Select Users For Which Assertions Will Be Generated](#) (see page 288)
- [Specify Name Identifiers for SAML 2.0 Assertions](#) (see page 290)
- [Configure Required General Information](#) (see page 292)
- [Authentication Users with no SiteMinder Session \(SAML 2.0\)](#) (see page 296)
- [Configure Single Sign-on for SAML 2.0](#) (see page 298)
- [Configure the Authentication Scheme that Protects the Artifact Service](#) (see page 314)
- [Configure Attributes for Assertions \(optional\)](#) (see page 319)
- [Set Up Links at the IdP or SP to Initiate Single Sign-on](#) (see page 323)
- [Configure Single Logout \(optional\)](#) (see page 331)
- [Configure Identity Provider Discovery at the IdP](#) (see page 334)
- [Encrypt a NameID and an Assertion](#) (see page 336)
- [Request Processing with a Proxy Server at the IdP](#) (see page 337)
- [HTTP Error Handling at the IdP](#) (see page 339)
- [Customize a SAML Response Element \(optional\)](#) (see page 339)

Prerequisites for a SiteMinder Asserting Party

For SiteMinder to serve as the asserting party, the following conditions must be met:

- Install the Policy Server.
- Install one of the following:
 - The Web Agent and the Web Agent Option Pack. The Web Agent authenticates users and establishes a SiteMinder session. The Option Pack provides the Federation Web Services application.
 - The SPS federation gateway, which has an embedded Web Agent and has the Federation Web Services application deployed on the embedded Tomcat web server.
- A SAML consumer is set up within the federated network.

The assertions generated by the producer are sent to the consumer. An application at the consumer receives and interprets the assertion. The SAML Affiliate Agent and a system with the SiteMinder Web Agent Option Pack can act as SAML consumers.

Configuration Checklist at the Identity Provider

Identifying a Service Provider to an Identity Provider is a task you complete at the SAML 2.0 Identity Provider because the Identity Provider needs information about the Service Provider to generate an assertion for that entity. Therefore, you identify the Service Provider to the Identity Provider and define how the two entities will communicate to pass assertions and to satisfy profiles, such as Web single sign-on or single logout.

- Required Configuration Tasks
- Optional Configuration Tasks

Tips:

- Certain parameter values at the Identity Provider and Service Provider must match for the configuration to work. A list of those parameters can be found in [Configuration Settings that Must Use the Same Values](#) (see page 471).
- To ensure you are using the correct URLs for the Federation Web Services servlets, a list of URLs can be found in [Federation Web Services URLs Used in SiteMinder Configuration](#) (see page 477).

How to Configure a SiteMinder Identity Provider

SiteMinder, as an Identity Provider generates assertions for its business partners, the Service Providers. To establish a federated partnership, the Identity Provider needs information about each partner. Create a Service Provider object for each partner and define how the two entities communicate to pass assertions and to satisfy profiles, such as single sign-on.

To configure a SiteMinder Identity Provider

1. Create a Service Provider object.
2. Add the Service Provider to an affiliate domain.
3. Specify the general identifying information for the Service Provider.
4. Select users from a user store. The Identity Provider generates assertions for these users.
5. Specify the Name ID.
6. Configure a single sign-on (SSO) profile.

You can save a Service Provider entity without configuring a complete SSO profile. However, you cannot pass an assertion to the Service Provider without completing the SSO configuration.

7. Configure signing and encryption for requests and responses.
8. Complete optional configuration tasks.

Tips:

- Certain parameter values at the Identity Provider and Service Provider must match for the configuration to work. A list of those parameters can be found in [Configuration Settings that Must Use the Same Values](#) (see page 471).
- Use the correct URLs for the Federation Web Services servlets. A list of URLs can be found in [Federation Web Services URLs Used in SiteMinder Configuration](#) (see page 477)

Optional Configuration Tasks for Identifying a Service Provider

The following optional tasks are for identifying a Service Provider:

- Configure IP address restrictions to limit the addresses that are used to access Service Providers.
- Configure time restrictions for Service Provider operations.
- Enable enhanced client or proxy profile.
- [Configure attributes](#) (see page 319) for inclusion in assertions.
- Configure single logout (SLO).
- [Configure the Identity Provider Discovery profile](#) (see page 334).
- [Encrypt the Name ID](#) (see page 336) in the assertion and/or the entire assertion.
- Sign the assertion and/or the entire assertion response.
- Sign the artifact resolve message and/or the artifact response.
- Customize a SAML assertion response using the Assertion Generator plug-in.

Add a SAML 2.0 Service Provider to an Affiliate Domain

To identify a Service Provider as a available consumer of SiteMinder-generated assertions, add the Service Provider to an affiliate domain configured at the Identity Provider's Policy Server. You then define the Service Provider's configuration so that the Identity Provider can issue assertions for it.

To add a Service Provider to an affiliate domain

1. Log into the FSS Administrative UI.
2. Display the list of domains.
3. Expand the AffiliateDomain object to reveal the SAML Service Providers object.

4. Select SAML Service Providers.
5. From the menu bar, select Edit, Create Service Provider.
The SAML Service Provider Properties dialog opens.
6. Fill in the following fields at the top of the dialog:
 - Name
 - Description
 - Authentication URL
 - Use Secure URL
 - Application URL

Note: Click Help for a description of fields, controls, and their respective requirements.
7. Check Enabled to enable the Identity Provider to recognize the Service Provider you have identified.

More Information:

[Enable the Assertion Generator Plug-in \(SAML 2.0\)](#) (see page 341)

Select Users For Which Assertions Will Be Generated

When you configure a Service Provider, you include a list of users and groups for which the Assertion Generator will generate SAML assertions. You may only add users and groups from directories that are in an affiliate domain.

To specify users and groups that have access to Service Provider resources

1. Log into the FSS Administrative UI.
2. Access the SAML Service Provider Properties dialog box and select the Users tab.
If the associated affiliate domain contains more than one user directory, the directories appear as subordinate tabs on the Users tab.
3. Click the Add/Remove button.
The Users/Groups dialog box opens.

4. To add users, select an entry from the Available Members list and click the Left Arrow button, which points to the Current Members list.

The opposite procedure removes users from the Current Members list.

You can select multiple entries by holding the CTRL or SHIFT key and clicking entries in one of the Members lists. When you select multiple entries and click one of the Arrow buttons, the FSS Administrative UI moves all of the selected entries.

Individual users are not displayed automatically. However, you can use the Search utility to find a specific user within one of the listed groups. Different types of user directories must be searched differently.

5. Click OK to save your changes.

Exclude a User or Group from Service Provider Access

You can exclude users or groups of users from obtaining an assertion. This is useful if you have a large user group that should have access to a Service Provider, but you there is a small subset of this group that you want to exclude.

To exclude a user or group from access to an Service Provider's resources

1. In the Users/Groups dialog box, select a user or group from the Current Members list.
2. To exclude the selected user or group, click Exclude.

The symbol to the left of the user or group in the Current Members list changes to indicate that the user or group is excluded from the Service Provider.

3. Click OK.

Allow Nested LDAP Groups Service Provider Access

LDAP user directories may contain groups nested in other groups. In complex directories, large amounts of user information may be organized in a nested hierarchy.

If you enable a Service Provider to search for users in nested groups, any subset group from a larger group that you add to a policy is searched by the Policy Server. If you do not enable nested groups, the Policy Server only searches the single group you specify for the Service Provider.

To allow the Service Provider to search nested groups in an LDAP user directory:

1. From the Users tab, select the Allow Nested Groups check box to enable nested groups searching for the Service Provider.
2. If the associated affiliate domain contains more than one user directory, the directories appear as tabs on the User tab.

Add Users by Manual Entry for Access to a Service Provider

From the Users/Groups dialog box, you can use the Manual Entry option to add users who can access the Service Provider resources.

To add a user by manual entry:

1. In the Manual Entry group box, do one of the following:

- For LDAP directories, enter a valid DN in the Entry field.

For each DN specified in the Entry field, you can select an action from the Action drop down list, as follows:

Search Users--the LDAP search is limited to matches in user entries.

Search Groups--the LDAP search is limited to matches in group entries.

Search Organizations--the LDAP search is limited to matches in organization entries.

Search Any Entry--the LDAP search is limited to matches in user, group, and organization entries.

Validate DN--the LDAP search locates this DN in the directory.

- For Microsoft SQL Server, Oracle and WinNT directories, enter a user name in the Manual Entry field.

For an Microsoft SQL Server or Oracle, you can enter a SQL query instead. For example:

```
SELECT NAME FROM EMPLOYEE WHERE JOB ='MGR';
```

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

Note: For an LDAP directory, you can enter all in the Manual Entry field to add all directory entries to the Service Provider.

2. Click Add to Current Members.

The FSS Administrative UI adds the user or query to the Current Members list.

3. Click OK to save your changes.

Specify Name Identifiers for SAML 2.0 Assertions

A name ID names a user in an assertion in a unique way. The value you configure in the FSS Administrative UI will be included in the assertion sent to the Service Provider.

The format of the name ID establishes the type of content used for the ID. For example, the format might be the User DN, in which case the content would be a uid.

You can encrypt a Name ID; however, if you are using single sign-on with the artifact binding, encrypting a NameID along with other data in an assertion increases the size of the assertion.

More Information:

[Encrypt a NameID and an Assertion](#) (see page 336)

[Allow the Identity Provider to Assign a Value for the NameID](#) (see page 310)

Configure a Name ID

To configure a name ID

1. Log in to the FSS Administrative UI and access the Service Provider entry you want to configure.
2. Select the Name IDs tab on the SAML Service Providers dialog box.
3. Select the Name ID Format.

For a description of each format, see Section 8.3 of the *Assertions and Protocols for the OASIS Security Assertion Markup Language (SAML) V2.0* specification (sstc-saml-core-2.0-cd-04.pdf).

4. Choose the Name ID Type from the following options:
 - Static value
 - User attribute
 - DN attribute (with or without nested groups)

The contents of the Name ID Fields group box change according to the Name ID Type selected.

5. Complete the fields for the selected Name ID Type.

Note: If you configure Name IDs, do not select an affiliation in the SAML Affiliation field. Name IDs and affiliations are mutually exclusive.

Configure a SAML 2.0 Affiliation (Optional)

If you configure an affiliation, all the Name ID settings are disabled; only the affiliation settings will be relevant.

An affiliation can be a group of Service Providers or Identity Providers. In this case, we are referring to an affiliation of Service Providers. If there is an established federated relationship between an Identity Provider and a Service Provider and that Service Provider is part of an affiliation, then the name identifier for that Service Provider will be the same identifier for all Service Providers in the affiliation.

To select an affiliation, choose an affiliation from the drop-down list for this Service Provider.

Configure Required General Information

Select the General tab to configure required items, such as the ID of the Service Provider and Identity Provider, or the SAML version being used for generating assertions.

To configure the general settings:

1. Log in to the FSS Administrative UI.
2. Open the SAML Service Provider Properties dialog.
3. Select the General tab and fill in values for the following required fields:

SP ID

Specifies a URI that uniquely identifies the Service Provider, such as sp.example.com.

IdP ID

Specifies a URI that uniquely identifies the Identity Provider, such as idp.ca.com. The IdP ID becomes the Issuer field in the assertion.

Skew Time

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

For single sign-on, the value of the Skew Time and the single sign-on validity duration (Validity Duration field on the SSO tab) determine how long an assertion is valid. Review how the [assertion validity is calculated](#) (see page 298) to understand more about the skew time.

For single logout, the values of the Skew Time and the SLO validity duration (Validity Duration field on the SLO tab) determine the total time that the single logout request is valid. Review how the [single logout request validity](#) (see page 332) is calculated to understand more about the skew time.

Set a Password for SAML Artifact Back Channel Authentication

If you use the HTTP-Artifact binding for SAML 2.0 single sign-on, the assertion is sent from the Identity Provider, across a secure back channel, to the Service Provider. You need to configure a password for the Service Provider to be granted access to the Artifact Resolution Service, which will resolve the artifact and retrieve the assertion.

Note: The password is only relevant if you use Basic or Basic over SSL as the authentication method across the back channel; however, you must configure a password regardless of which authentication method you plan to use.

To configure a password for HTTP-Artifact binding:

1. Open the SAML Service Provider Properties dialog.
2. On the General tab, click Configure Backchannel Authentication.

The Backchannel Properties dialog opens.

Note: The Configure Backchannel Authentication button is only active if you select HTTP-Artifact on the SSO tab.

3. Enter a value for the following fields:
 - Password
 - Confirm Password
4. Click OK.

You return to the SAML Service Provider Properties dialog.

WebLogic Configuration Required for Back Channel Authentication

At the Identity Provider, the Web Agent Option Pack can be installed on a WebLogic 9.2.x application server. For basic authentication across the back channel to work with this server, modify the WebLogic config.xml file.

In the WebLogic config.xml file for the application domain, set the `<enforce-valid-basic-auth-credentials>` within the `<security-configuration>` element as follows:
`<enforce-valid-basic-auth-credentials>>false</enforce-valid-basic-auth-credentials>`

Determine Digital Signing Options

SiteMinder can use a private key/certificate pair to perform various digital signing tasks for federated communication. The private key can sign the following:

- Assertions
- SAML responses
- Artifact responses
- Single logout requests and responses

For single logout, the side that initiates the logout signs the request, and the side receiving the request validates the signature. Conversely, the receiving side must sign the SLO response and the initiator must validate the response signature.

- Attribute responses (for authorizations based on user attributes)

Prior to any transaction involving signing, the partner responsible for signing gives the certificate (public key) associated with the private key to the partner that verifies the signature. This exchange is done in an independent communication from the federated transaction.

When a SiteMinder IdP sends an assertion to an SP, it includes the certificate in the assertion, by default. However, the SP uses the certificate that it stores at its site to verify the signature.

The configuration options for digital signing include:

- The signature alias setting
- The signature algorithm (RSAwithSHA1 or RSAwithSHA256)
- HTTP-Artifact assertion, SAML response, and artifact response options
- HTTP-POST assertion and SAML response options

To specify signing options from the General or SSO tab

1. Open the SAML Service Provider Properties dialog.
2. Select the General or SSO tab.
3. Select Signing Options.

The Signing Options dialog opens.

Complete the fields in the Signing Options dialog.

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

Validate Signed AuthnRequests and SLO Requests/Responses

By default, signature processing is enabled because it is required by the SAML 2.0 specification; therefore, it *must* be enabled in a production environment. SAML 2.0 POST responses and single logout requests are always signed by SiteMinder; signing does not require configuration using the FSS Administrative UI.

For signing, the only setup required is that you have to add the private key and certificate of the authority responsible for signing to the smkeydatabase.

Important! For debugging purposes only, you can temporarily disable all signature processing (both signing and verification of signatures) by checking the **Disable Signature Processing** option.

To validate signatures of AuthnRequests from a Service Provider, or single logout requests and responses, there are configuration steps in the FSS Administrative UI and the smkeydatabase.

To set-up validation:

1. Add the public key to the Identity Provider's smkeydatabase.

The public key must correspond to the private key and certificate that the Service Provider used to do the signing.

Note: To see updates to the smkeydatabase immediately, restart the Policy Server. Otherwise, the database is updated based on the frequency you configured in the smkeydatabase.properties file.

2. In the FSS Administrative UI, select one or both of the following check boxes:

- **Require Signed AuthnRequests** (on the SSO tab)

If you select this check box, the Identity Provider will require a signed authnrequest and then validate the signature of the request. If the authnrequest is not signed, it will be rejected.

Important: If you sign AuthnRequests, no unsolicited responses can be sent from the Identity Provider.

- **HTTP-Redirect** (on the SLO tab)

If you select this check box, the Identity Provider will validate the signature of the SLO request and response.

3. Complete the Issuer DN and Serial Number fields on the General tab.

The Issuer DN and Serial Number fields become active only after the **Require Signed AuthnRequests** or the **HTTP-Redirect** check box is selected. The values you enter for these fields should match the public key in the smkeydatabase that corresponds to the private key and certificate of the authority that signed the requests. We recommend you open a command window and enter the command `smkeytool -lc` to list the certificates and view the DN to ensure that you enter a matching value.

Authentication Users with no SiteMinder Session (SAML 2.0)

When you add a Service Provider to an affiliate domain, one of the parameters you are required to set is the AuthenticationURL parameter.

The file that the Authentication URL points to is the redirect.jsp file. This file is installed at the Identity Provider site where you install the Web Agent Option Pack or the SPS federation gateway. The redirect.jsp file must be protected by a SiteMinder policy so that an authentication challenge is presented to users who request a protected Service Provider resource but do not have a SiteMinder session.

A SiteMinder session is required for the following bindings:

- For users requesting a protected Service Provider resource
If you configure single sign-on using an HTTP artifact binding, a persistent session is needed to store SAML assertions in the session server.
- For single sign-on using an HTTP POST binding
A user must have a session, but it does not have to be a persistent session because assertions are delivered directly to the Service Provider site through the user's browser. The assertions do not have to be stored in the session server.
- For single logout
If you enable single logout, a persistent session is required. When a user first requests a Service Provider resource, the session established at that time must be stored in the session server so that the necessary session information is available when a single logout is later executed.

After a user is authenticated and successfully accesses the redirect.jsp file, a session is established. The redirect.jsp file redirects the user back to the Identity Provider Web Agent or the SPS federation gateway so that the request can be processed and delivered to the SAML assertion for the user.

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

- Web Agent Option Pack installed on the same system as the Web Agent
- Application server with a Web Agent installed on a Web server proxy
- Application server protected by an Application Server Agent
- SPS federation gateway installed at the Identity Provider

Create a Policy to Protect the Authentication URL

To create a policy to protect the AuthenticationURL

1. Open the FSS Administrative UI.
2. From the System tab, create Web Agents to bind to the realms that you define for the producer-side Web Server. You can assign unique Agent names for the Web Server and the Federation Web Services or use the same Agent name for both.
3. Create a policy domain for the users who should be challenged when they try to access a consumer resource.
4. From the Users tab, select the users that should have access to the resources that are part of the policy domain.
5. Define a realm for the policy domain with the following values:
 - a. Agent: select the Agent for the producer Web Server
 - b. Resource Filter:

Web Agents v5.x QMR 4 and later, and SPS federation gateway enter:
`/siteminderagent/redirectjsp/`

Web Agents v5.x QMR 1, 2, or 3, enter:
`/affwebservices/redirectjsp/`

The resource filter `/siteminderagent/redirectjsp/` is an alias, set up automatically by FWS. It is a reference to the following:

 - For a Web Agent:
`web_agent_home/affwebservices/redirectjsp`
 - For the SPS federation gateway:
`sps_home/secure-proxy/Tomcat/webapps/affwebservices/redirectjsp`
 - c. For the remaining settings, accept the defaults or modify as needed.
6. For SAML artifact only, select the Session tab and check the Persistent Session check box.

To enable single sign-on using the SAML artifact profile from a realm at the producer to a realm at the consumer, configure a persistent session for the producer realm. If you do not configure a persistent session, the user cannot access consumer resources.
7. Click OK to save the realm.
8. Create a rule for the realm. In the Resource field, accept the default value, the asterisk (*), to protect all resources for the realm.

9. Create a policy for the producer Web Server that includes the rule created in the previous step.
10. Complete the task in [Select Users for Which Assertions Will Be Generated](#) (see page 228).

Configure Single Sign-on for SAML 2.0

The Service Provider and the Identity Provider exchange user information, session information and Identity Provider information in an assertion document. When you configure single sign-on at the SAML 2.0 Identity Provider, you determine how the Identity Provider delivers an assertion to a Service Provider.

The sections that follow and the Help in the FSS Administrative UI provide guidance for configuring various settings.

To configure single sign-on at the Identity Provider

1. Log on to the FSS Administrative UI.
2. Select a Service Provider entry.
3. Right-click the entry to access the SAML Service Provider Properties dialog for the selected Service Provider.
4. Select the SSO tab.
5. Complete the fields on the SSO tab.
Refer to the SAML 2.0 Service Provider reference for field descriptions.
6. Click OK to save your changes.

You have now defined the single sign-on settings at the Identity Provider that it will use to communicate with the Service Provider.

Assertion Validity for Single Sign-on

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

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

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

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

NotBefore=IssueInstant - Skew Time

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

NotOnOrAfter=Validity Duration + Skew Time + IssueInstant

Times are relative to GMT.

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

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

Calculating Assertion Validity with SiteMinder at Both Sides of the Partnership

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

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

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

Note: For Federation Security Services, the Validity Duration is only set at the asserting party.

Example

Asserting Party

The values at the asserting party are as follows:

IssueInstant=5:00PM

Validity Duration=60 seconds

Skew Time = 60 seconds

NotBefore = 4:59PM

NotOnOrAfter=5:02PM

Relying Party

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

Skew Time = 180 seconds (3 minutes)

NotBefore = 4:56PM

NotOnOrAfter=5:05PM

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

$120 \text{ seconds } (2 \times 60) + 60 \text{ seconds } + 360 \text{ seconds } (2 \times 180) = 540 \text{ seconds } (9 \text{ minutes}).$

Configure an Assertion for One Time Use

SiteMinder can generate an assertion that is intended for one time use by the relying party, requiring the relying party to request a new assertion each time it needs one. Restricting an assertion to one use helps ensure that authentication decisions are based on current information.

To generate an assertion with a one time use condition

1. Log on to the FSS Administrative UI.
2. Select the Service Provider you want to modify or create one.
3. Navigate to the Advanced tab.
4. Select the Set OneTimeUse Condition check box.
5. Click OK.

The asserting party can now generate an assertion that includes the condition element for its one time use.

Customize the Session Duration in the Assertion

When the SiteMinder IdP sends an assertion, by default it includes the SessionNotOnOrAfter parameter in the Authentication statement of the assertion. A third-party SP can use the value of the SessionNotOnOrAfter to set its own timeout values. The timeout values determine when a user session becomes invalid, which sends the user to reauthenticate at the IdP.

Important! If SiteMinder is acting as an SP, it ignores the SessionNotOnOrAfter value. Instead, a SiteMinder SP sets session timeouts based on the realm timeout that corresponds to the configured SAML authentication scheme that protects the target resource.

Note: The SessionNotOnOrAfter parameter is different than the NotOnOrAfter parameter used to determine assertion validity and skew time.

To customize the SessionNotOnOrAfter parameter

1. Log on to the UI.
2. Select the Service Provider entry that you want to modify.
3. Navigate to the Advanced tab.
4. Select the Customize Validity duration in the Advanced SSO Configuration section of the dialog.

The Customize Validity duration dialog displays.

5. Select a value for the SP Session Validity Duration. The value that you enter is the value of the SessionNotOnOrAfter parameter in the assertion.

The options are:

Use Assertion Validity

Calculates the SessionNotOnOrAfter value that is based on the assertion validity duration.

Omit

Instructs the IdP not to include the SessionNotOnOrAfter parameter in the assertion.

IDP Session

Calculates the SessionNotOnOrAfter value that is based on the IdP session timeout. The timeout is configured in the IdP realm for the authentication URL. Using this option can synchronize the IdP and SP session timeout values.

Custom

Lets you specify a custom value for the SessionNotOnOrAfter parameter in the assertion. If you select this option, enter a time in the Customize Assertion Session Duration field.

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

6. Click OK to save the changes.

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

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

To grant access:

- [Add the Web Agent](#) (see page 302) that protects the FWS application to the agent group FederationWebServicesAgentGroup.
- [Add relying partners as users](#) (see page 303) who are permitted to access the specific service.

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

Add a Web Agent to the Federation Agent Group

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

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

Follow these steps:

1. Log in to the Administrative UI.
2. Select Infrastructure, Agents, Create Agent.
3. Specify the name of the Web Agent in your deployment. Click Submit.
4. Select Infrastructure, Agent Groups.

5. Select the FederationWebServicesAgentGroup entry.
The Agent Groups dialog opens.
6. Click Add/Remove and the Agent Group Members dialog opens.
7. Move the web agent from the Available Members list to the Selected Members list.
8. Click OK to return to the Agent Groups dialog.
9. Click Submit then click Close to return to the main page.

Add Relying Partners to the FWS Policy for Obtaining Assertions

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

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

SAML 1.x

FederationWSAssertionRetrievalServicePolicy

SAML 2.0

SAML2FWSArtifactResolutionServicePolicy

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

Grant access for these policies to any relevant relying partners.

Follow these steps:

1. In the Administrative UI, navigate to Policies, Domain, Domain Policies.
A list of domain policies displays.

2. Select the policy for the SAML profile:

SAML 1.x

FederationWSAssertionRetrievalServicePolicy

SAML 2.0

SAML2FWSArtifactResolutionServicePolicy

The Domain Policies page opens.

3. Click Modify to change the policy.
4. Select the Users tab.

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

SAML 1.x

FederationWSCustomUserStore

SAML 2.0

SAML2FederationCustomUserStore

The User/Groups page opens.

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

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

You return to the User Directories list.

7. Click Submit.

You return to the policies list.

Verify Basic Protection of the Assertion Retrieval Service

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

Follow these steps:

1. Open a web browser.

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

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

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

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

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

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

Define Indexed Endpoints for Different Single Sign-on Bindings

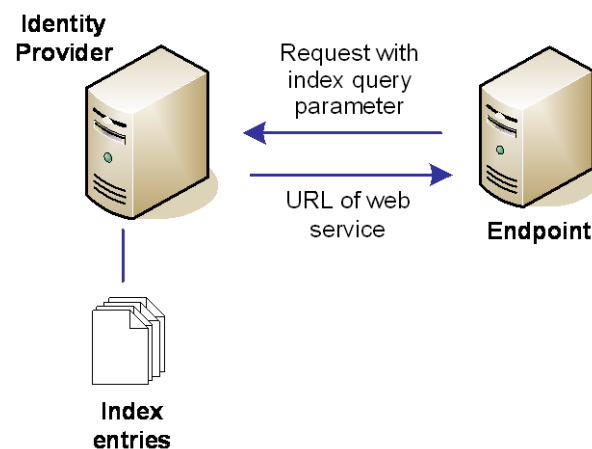
You can configure indexed endpoints for federated communication. An indexed endpoint is the site where assertions are consumed. In the context of SiteMinder, this endpoint is the Service Provider where the Assertion Consumer Service resides.

Each endpoint you configure is assigned a unique index value, instead of a single, explicit reference to an Assertion Consumer Service URL. The assigned index is added to the assertion request that the Service Provider sends to the Identity Provider.

You can configure indexed endpoints for a SiteMinder Service Provider that has a federated relationship with a third-party Identity Provider that supports indexed endpoints. You can also configure different protocol bindings (artifact, POST) for the Assertion Consumer Service by assigning more than one endpoint to the service.

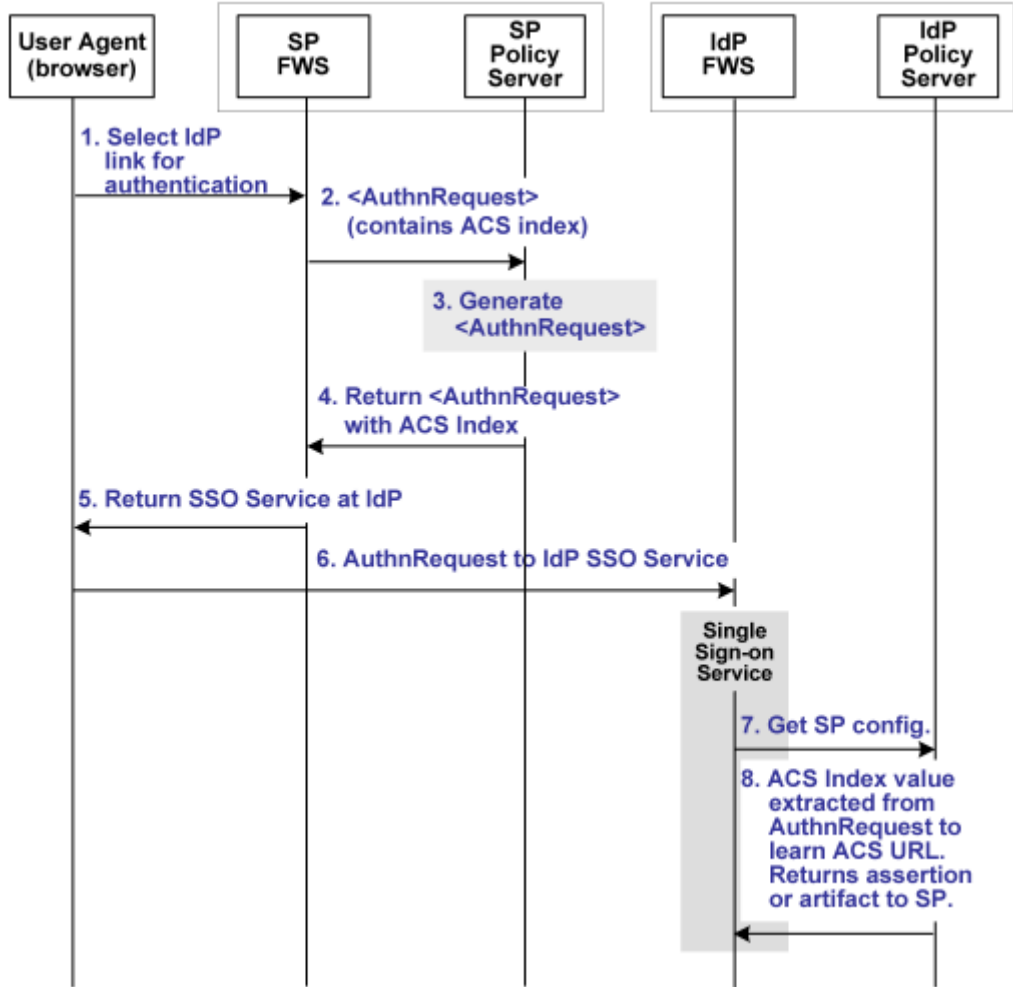
Note: If your network contains different SiteMinder versions, for example, the Service Provider is r12.0 SP2 and the Identity Provider is r12.0 SP3, you cannot configure indexed endpoints. Configure only one Assertion Consumer Service for both HTTP bindings.

The following figure shows a network that benefits from indexed endpoints.



Indexed Endpoints Flow Diagram

The following illustration shows how single sign-on works using an indexed endpoint.



Note: The Web Agent Option Pack or the SPS federation gateway can provide the FWS functionality.

Using indexed endpoints, the sequence of events is as follows:

1. The user selects a link to authenticate with a specific IdP. The link contains the IdP ID and AssertionConsumerServiceIndex query parameters index as query parameters because the index feature is enabled.
2. The SP Federation Web Services (FWS) application asks for an AuthnRequest from its local Policy Server. The request that it sends includes the IdP ID and optionally, the AssertionConsumerServiceIndex and ForceAuthn query parameters.

A protocol binding is not part of the request because the ACS Index and the Protocol Binding parameters are mutually exclusive. The AssertionConsumerServiceIndex is already associated with a binding so there is no need to specify a Protocol Binding value. If the protocol binding and the AssertionConsumerServiceIndex are passed as query parameters, the local Policy Server responds with an error denying the request.

3. The AuthnRequest service extracts the IdP information from the SP Policy Server and generates the AuthnRequest message, which includes the AssertionConsumerServiceIndex. Because the AssertionConsumerServiceIndex is one of the query parameters, its value is verified against the IdP from an IdP descriptor document. This document is previously sent from the IdP to the SP.

The AuthnRequest service reacts as follows:

- If the index for the artifact binding is set in the IdP metadata, this index is compared to the AssertionConsumerServiceIndex value. If the values match, the index value remains part of the AuthnRequest. If the index values do not match, the IdP metadata is verified. The AssertionConsumerServiceIndex must correspond to the POST binding.
 - If the index corresponding to the HTTP-POST binding, this index value is again compared with the AssertionConsumerServiceIndex in the AuthnRequest. If the value of the AssertionConsumerServiceIndex parameter does not match the POST binding, the AuthnRequest service generates an error. The error states that the AssertionConsumerServiceIndex does not match the index in the IdP metadata.
4. Assuming that the IdP metadata index and AssertionConsumerServiceIndex values match, the SP Policy Server generates the AuthnRequest.
 5. The SP Policy Server returns the AuthnRequest in an HTTP-redirect binding.
 6. The the SP FWS application redirects the AuthnRequest to the single sign-on service at the IdP. The SP knows the URL of the single sign-on service because the URL is part of the configuration information in the AuthnRequest.

7. The browser requests the single sign-on service.
8. The single sign-on service extracts the AssertionConsumerServiceIndex value from the AuthnRequest. The service determines the Assertion Consumer Service URL using the AssertionConsumerServiceIndex. If the Index is not in the metadata, the service generates an error. The error message indicates that an invalid AssertionConsumerServiceIndex is in the AuthnRequest message.

The Assertion Consumer URL to send the assertion or artifact to the SP, depending on the single sign-on profile in use.

Note: If the AssertionConsumerServiceIndex parameter is not in the AuthnRequest, the value of the Assertion Consumer Service and the corresponding binding are used by default.

Configure Indexed Endpoints for the Assertion Consumer Service

When the single sign-on service extracts an ACS Index value from a Service Provider's AuthnRequest, it compares the index value to its list of index entries and determines the Assertion Consumer Service URL associated with that index value. The single sign-on service then knows where to send the assertion or artifact, depending on the binding associated with the index value.

To configure index entries at the Identity Provider

1. Log in to the FSS Administrative UI.
2. Display the list of domains and from the Affiliate domain, select the Service Provider you want to configure.

The SAML Service Provider Properties dialog opens.

3. Select the SSO tab.
4. Click the ellipses button at the end of the Assertion Consumer Service field.

The Assertion Consumer Service dialog opens.

5. Click on Add to define an index entry.

The Add Assertion Consumer Service dialog opens.

6. Complete the following required fields:

- Index
- Binding
- Assertion Consumer Service URL

Note: You can use different index values assigned to the same Assertion Consumer Service URL.

7. Click OK to save your changes.

Note: Remember to configure index entries in the SAML 2.0 authentication scheme at the Service Provider.

Enforce the Authentication Scheme Protection Level for SSO

When a user requests a federated resource, they must have a SiteMinder session. If a user does not have a SiteMinder session, the user is redirected to the Authentication URL to establish a session. The authentication scheme protecting the Authentication URL is configured with a particular protection level. This protection level must be the same or greater than the authentication level you configure for the SAML Service Provider configuration.

If the protection level for the Authentication URL is less than the Authentication Level set in the Administrative UI, SiteMinder does not generate an assertion.

Determine Digital Signing Options

SiteMinder can use a private key/certificate pair to perform various digital signing tasks for federated communication. The private key can sign the following:

- Assertions
- SAML responses
- Artifact responses
- Single logout requests and responses

For single logout, the side that initiates the logout signs the request, and the side receiving the request validates the signature. Conversely, the receiving side must sign the SLO response and the initiator must validate the response signature.

- Attribute responses (for authorizations based on user attributes)

Prior to any transaction involving signing, the partner responsible for signing gives the certificate (public key) associated with the private key to the partner that verifies the signature. This exchange is done in an independent communication from the federated transaction.

When a SiteMinder IdP sends an assertion to an SP, it includes the certificate in the assertion, by default. However, the SP uses the certificate that it stores at its site to verify the signature.

The configuration options for digital signing include:

- The signature alias setting
- The signature algorithm (RSAwithSHA1 or RSAwithSHA256)
- HTTP-Artifact assertion, SAML response, and artifact response options
- HTTP-POST assertion and SAML response options

To specify signing options from the General or SSO tab

1. Open the SAML Service Provider Properties dialog.
2. Select the General or SSO tab.
3. Select Signing Options.

The Signing Options dialog opens.

Complete the fields in the Signing Options dialog.

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

Allow the Identity Provider to Assign a Value for the NameID

As part of a single sign-on request, a Service Provider may request a particular user attribute to be included in the assertion; however, the value of the required attribute may not be available in the user record at the Identity Provider.

If the Service Provider's request includes the Allow/Create attribute and the Identity Provider is configured to create a new identifier, the Policy Server at the Identity Provider will generate a unique value as part of the NameID. This value is then included in the assertion that is sent back to the Service Provider.

When the Service Provider receives the assertion, the SAML 2.0 authentication scheme processes the response, performs a user lookup in its local user store, and assuming the user record is located, the user is granted access.

Enable the Creation of a Name Identifier

To enable the creation of a new name identifier for single sign-on

1. Log in to the FSS Administrative UI.
2. Display the list of domains and from the Affiliate domain, select an existing Service Provider or create a new Service Provider.

The SAML Service Provider Properties dialog opens.

3. Select the SSO tab.
4. Check the Allow Creation of New Identifier check box.
5. Click OK.

Configure IP Address Restrictions for Service Providers (optional)

The FSS Administrative UI allows you to specify an IP address, range of IP addresses, or a subnet mask of the Web server on which a user's browser must be running for the user to access a Service Provider. If IP addresses have been specified for a Service Provider, only users who access the Service Provider from the appropriate IP addresses will be accepted by the Service Provider.

To specify IP addresses

1. Log in to the FSS Administrative UI and select the Service Provider you want to configure.
2. Open the SAML Service Provider Properties dialog box.
3. Select the SSO tab, then click on Restrictions.
4. Click Add.

The Add an IP Address dialog box opens.

5. Select one of the following radio buttons to indicate the type of IP address value you are adding:

Note: If you do not know the IP address, but you have a domain name for the address, you can click on the DNS Lookup button to open the DNS Lookup dialog box. Enter a fully qualified host name in the Host Name field and click OK.

- **Single Host**--specifies a single IP address that hosts the user's browser. If you specify a single IP address, the Service Provider can only be accessed by users from the specified IP address.
 - **Host Name**--specifies a Web server using its host name. If you specify a host name, the Service Provider is only accessible to users who access it from the specified host.
 - **Subnet Mask**--specifies a subnet mask for a Web server. If you specify a subnet mask, the Service Provider is only accessible to users who access the Service Provider resources from the specified subnet mask. If you select this button, the Add An Address and Subnet Mask dialog opens. Use the Left and Right arrow buttons, or click and drag the slider bar to select a subnet mask.
 - **Range**--specifies IP address range. If you specify a range of IP addresses, the Service Provider only permits users who access the Service Provider resources from one of the IP addresses in the range of addresses. You enter a starting (FROM) and ending (TO) addresses to determine the range.
6. Click OK to save your configuration.

Configure Time Restrictions for Service Provider Availability (optional)

You can specify time restrictions that indicate a Service Provider's availability. When you add a time restriction, the Service Provider functions only during the period specified. If a user attempts to access a resource outside of that period, the Identity Provider does not produce SAML assertions.

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

To specify a time restriction:

1. Log in to the FSS Administrative UI and select the Service Provider you want to configure.
2. Open the SAML Service Provider Properties dialog box.
3. Select the SSO tab, then click on Restrictions.
4. In the Time Restrictions group box, click Set.

The Time dialog box opens. This dialog box is identical to the Time Restrictions dialog box used for rule objects.

5. Click OK.

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.

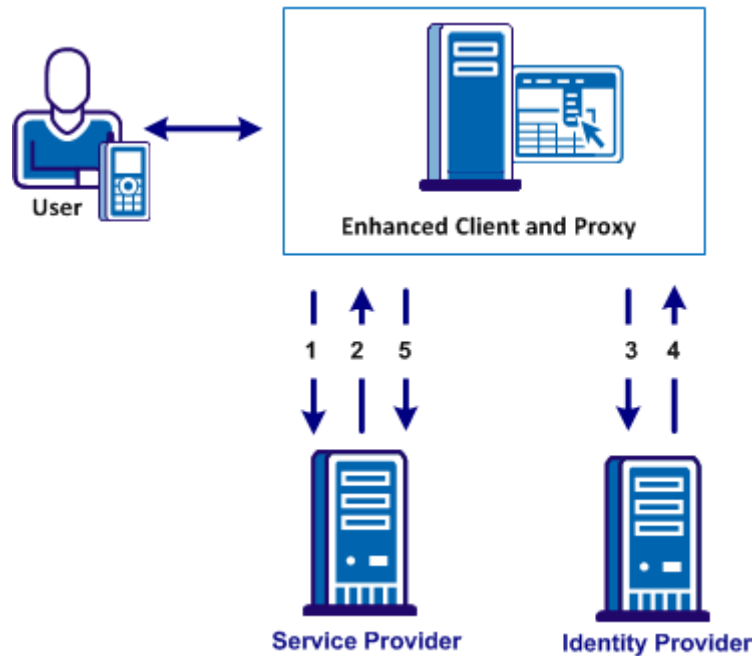
The ECP profile enables single sign-on when the Identity Provider and Service Provider cannot communicate directly. The ECP acts as the intermediary between the Service Provider and the Identity Provider.

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. SiteMinder 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 with SiteMinder, enable the feature at the Identity Provider and the Service Provider. The following procedure is for a SiteMinder Identity Provider.

Follow these steps:

1. Log in to the FSS Administrative UI at the Identity Provider.
2. Navigate to the SSO tab for the SAML Service Provider object you want to modify.
3. Complete the required single sign-on configuration settings in the dialog.
4. Select the Enhanced Client and Proxy Profile check box.
5. Click OK.

The SiteMinder Identity Provider can now process ECP calls.

Note: A single SAML Service Provider object can handle artifact, POST, SOAP, and PAOS bindings for single sign-on requests. SOAP and PAOS are the bindings for the ECP profile. The Identity Provider and Service Provider determine the binding being used based on the parameters in a request.

Configure the Authentication Scheme that Protects the Artifact Service

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

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

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

You can select one of the following authentication schemes:

- [Basic](#) (see page 315)
- Basic over SSL
- [X.509 client certificate](#) (see page 238)

Basic Authentication to Protect the Service that Retrieves Assertions

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

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

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

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

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

Follow these steps: for the SAML 2.0 Artifact Resolution Service

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

Basic over SSL to Protect the Assertion Retrieval Service

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

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

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

Client Certificate Auth to Protect the Service that Retrieves Assertion

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

To use a client certificate authentication scheme:

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

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

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

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

More Information:

[Configure the Client Certificate Authentication at the Relying Party](#) (see page 381)

Create the Policy to Protect the Retrieval Service

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

Follow these steps:

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

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

```
uid=CompanyA, ou=Development, o=CA
```

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

2. Add the configured user directory to the FederationWebServicesDomain.
3. Create a certificate mapping entry.

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

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

4. Configure an X509 Client Certificate authentication scheme.
5. Create a realm under the FederationWebServicesDomain containing the following entries:

Name

any_name

Example: cert assertion retrieval

Agent

FederationWebServicesAgentGroup

Resource Filter

/affwebservices/certassertionretriever (SAML 1.x)

/affwebservices/saml2certartifactresolution (SAML 2.0)

Authentication Scheme

Client certificate authentication scheme created in the previous step.

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

Name

any_name

Example: cert assertion retrieval rule

Resource

*

Web Agent Actions

GET, POST, PUT

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

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

Create a response with the following values:

Name

any_name

Attribute

WebAgent-HTTP-Header-Variable

Attribute Kind

User Attribute

Variable Name

consumer_name

Attribute Name

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

Example: uid=CompanyA.

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

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

Name

any_name

User

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

Rule

rule_created_earlier_in_this_procedure

Response

response_created_earlier_in_this_procedure

The policy to protect the artifact resolution service is complete.

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

SAML 1.x: [Enable client certificate authentication](#) (see page 275) for the Assertion Retrieval Service

SAML 2.0: [Enable client certificate authentication](#) (see page 275) for the Artifact Resolution Service

Configure Attributes for Assertions (optional)

Attributes can provide information about a user requesting access to a Service Provider resource. An attribute statement passes user attributes, DN attributes, or static data from the Identity Provider to the Service Provider in a SAML assertion. Any configured attributes are included in the assertion in one <AttributeStatement> element or the <EncryptedAttribute> element in the assertion.

Note: Attribute statements are not required in an assertion.

Servlets, web applications, or other custom applications use attributes to display customized content or enable other custom features. When used with web applications, attributes can implement fine-grained access control by limiting what a user can do at the Service Provider. For example, you can send an attribute variable named Authorized Amount set to a maximum dollar amount. The amount is the limit that the user can spend at the Service Provider.

Attributes take the form of name/value pairs. When the Service Provider receives the assertion, it takes the attribute values and makes them available to applications.

Attributes can be made available as HTTP Headers or HTTP Cookies.

The HTTP headers and HTTP cookies have size restrictions that assertion attributes cannot exceed. The size restrictions are as follows:

- For HTTP headers, SiteMinder can send an attribute in a header up to the web server size limit for a header. Only one assertion attribute per header is allowed. See the documentation for your web server to determine the header size limit.
- For HTTP cookies, SiteMinder can send a cookie up to the size limit for a cookie. Each assertion attribute is sent as its own cookie. The cookie size limit is browser-specific, and that limit is for all attributes being passed to the application, not for each attribute. See the documentation for your web browser to determine the cookie size limit.

Configure attributes in the Attributes tab of the Service Provider Properties dialog. Configuration involves choosing an Attribute Kind then filling in values for the variable name and attribute value.

Attributes for SSO and Attribute Query Requests

Indicate whether an attribute that you configure is for a single sign-on request, or for an attribute query request. The retrieval method that you configure determines the function of the attribute.

To use the same attribute for both services, create two attribute statements that use the same attribute name and variable. However, one attribute uses SSO as the retrieval method and one uses Attribute Service as the retrieval method.

Configure Attributes for SSO Assertions

Attributes can provide information about a user requesting access to a Service Provider resource. An attribute statement passes user attributes, DN attributes, or static data from the Identity Provider to the Service Provider in a SAML assertion.

To configure an attribute

1. In the Service Provider Properties dialog box, click on the Attributes tab.
2. Click Create.

The SAML Service Provider Attribute dialog box opens.

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

3. From the Attribute drop down list, select the name format identifier, as specified by the <NameFormat> attribute within the <Attribute> element of an assertion attribute statement. This value classifies the attribute name so that the Service Provider can interpret the name.

The options are:

unspecified

Determines how the name interpretation is left to your implementation

basic

Indicates that the name format must use acceptable values from the set of values belonging to the primitive type xs:Name.

URI

Indicates that the name format must follow the standards for a URI reference. How the URI is interpreted is specific to the application using the attribute value.

4. From the Attribute Setup tab, select one of the following radio buttons in the Attribute Kind group box. Your selection of the Attribute Kind radio button determines the available fields in the Attribute Fields group box.

Static

Returns data that remains constant.

User Attribute

Returns profile information from a user's entry in a user directory.

Note: For attributes from an LDAP user store, you can add multi-valued user attributes to an assertion. Review the Help for the Attribute Name field in this dialog for information about multi-valued attributes.

DN Attribute

Returns profile information from a directory object in an LDAP or ODBC user directory.

If you select the DN Attribute radio button, you can also select Allow Nested Groups. Selecting this check box allows SiteMinder to return an attribute from a group that is nested in another group. Nested groups often occur in complex LDAP deployments.

5. Configure the relevant fields in the Attribute Fields section of the dialog. The settings vary depending on the Attribute Kind selection. The options are:
 - Variable Name
 - Variable Value
 - Attribute Name
 - DN Spec

6. (Optional) if the attribute is retrieved from an LDAP user directory that contains nested groups (groups that contain other groups), and you want the Policy Server to retrieve DN attributes from the nested groups, select the Allow Nested Groups check box in the Attribute Kind group box.
7. (Optional) if you want the attribute values encrypted, select the Encrypted checkbox.
8. For the Retrieval Method, accept the default value SSO to ensure this attribute is used for single sign-on assertions and not for attribute assertions.
9. Click OK to save the changes.

Using a Script to Create A New Attribute

The Advanced tab of the SAML Service Provider Attribute dialog box contains the Script field. This field displays the script that SiteMinder generates based on your entries in the Attribute Setup tab. You can copy the contents of this field and paste them into the Script field for another response attribute.

Note: If you copy and paste the contents of the Script field for another attribute, you must select the appropriate radio button in the Attribute Kind group box of the Attribute Setup tab.

Specify the Maximum Length of Assertion Attributes

The maximum length for user assertion attributes is configurable. To modify the maximum length of assertion attributes, change the settings in the EntitlementGenerator.properties file.

Note: The property name in the file is specific to the protocol you are configuring.

Follow these steps:

1. On the system where the Policy Server is installed, navigate to `policy_server_home\config\properties\EntitlementGenerator.properties`.
2. Open the file in a text editor.

3. Adjust the maximum user attribute length for the protocols in use in your environment. The settings for each protocol are as follows:

WS-Federation

Property Name:

com.netegrity.assertiongenerator.wsfed.MaxUserAttributeLength

Property Type: Positive Integer value

Default Value: 1024

Description: Indicates the maximum attribute length for WS-FED assertion attributes.

SAML 1.x

Property Name:

com.netegrity.assertiongenerator.saml1.MaxUserAttributeLength

Property Type: Positive Integer value

Default Value: 1024

Description: Indicates the maximum attribute length for SAML1.1 assertion attributes.

SAML 2.0

Property Name:

com.netegrity.assertiongenerator.saml2.MaxUserAttributeLength

Property Type: Positive Integer value

Default Value: 1024

Description: Indicates the maximum attribute length for SAML2.0 assertion attributes

4. Restart the Policy Server after any change to these parameters.

Set Up Links at the IdP or SP to Initiate Single Sign-on

To initiate single sign-on, the user can begin at the Identity Provider or the Service Provider. You need to configure the appropriate links at each site to trigger single sign-on operation.

Identity Provider-initiated SSO (POST or artifact binding)

If a user visits the Identity Provider before going to the Service Provider, the Identity Provider must generate an unsolicited response. To initiate an unsolicited response, create a hard-coded link that generates an HTTP Get request that includes a query parameter with the Service Provider ID. The Identity Provider generates an assertion response for this ID. The Federation Web Service application and the Assertion Generator must accept the GET request.

A user clicks the link you establish to initiate the unsolicited response.

To specify the use of artifact or POST profile in the unsolicited response, the syntax for the unsolicited response link is:

```
http://idp_server:port/affwebservices/public/saml2sso?SPID=SP_ID&
ProtocolBinding=URI_for_binding
```

idp_server:port

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

SP_ID

Service Provider ID value.

URI_for_binding

Identifies the URI of the POST or Artifact binding for the ProtocolBinding element. The SAML 2.0 specification defines this URI.

Also specify the binding in the SAML Service Provider properties for the unsolicited response to work.

Note the following information:

- If there is no ProtocolBinding parameter in the link and only one binding in the Service Provider properties, the Service Provider uses the one binding.
- If the artifact and POST bindings are enabled in the Service Provider properties, POST is the default. Therefore, if you want to *only* use artifact binding, include the ProtocolBinding query parameter in the link.

Important! If you configure indexed endpoints for the Assertion Consumer Services, the ProtocolBinding query parameter overrides the binding for the Assertion Consumer Service.

More information:

[Unsolicited Response Query Parameters Used by a SiteMinder IdP](#) (see page 325)

Unsolicited Response Query Parameters Used by a SiteMinder IdP

An unsolicited response that initiates single sign-on from the IdP can include the following query parameters:

- SPID
- ProtocolBinding
- RelayState

SPID

(Required) Specifies the ID of the Service Provider where the Identity Provider sends the unsolicited response.

ProtocolBinding

Specifies the ProtocolBinding element in the unsolicited response. This element specifies the protocol used when sending the assertion response to the Service Provider. If the Service Provider is not configured to support the specified protocol binding, the request fails.

Required Use of the ProtocolBinding Query Parameter

Using the ProtocolBinding parameter is required *only* if the artifact and POST bindings are enabled in the Service Provider properties. If both profiles are enabled, use the query parameter only to use artifact binding.

- The URI for the artifact binding from the SAML 2.0 specification is:
urn:oasis:names:tc:SAML:2.0:bindings:HTTP-Artifact
- The URI for the POST binding from the SAML 2.0 specification is:
urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST

You do not need to set this parameter for HTTP-POST single sign-on.

Note: Do not HTTP-encode the query parameters.

Example: Unsolicited Response with ProtocolBinding

This link redirects the user to the Single Sign-on service. In this link is the Service Provider identity. The SPID query parameter indicates the identity. Additionally, the bindings query parameter indicates that the artifact binding is in use. After the user clicks this hard-coded link, they are redirected to the local Single Sign-on service.

```
http://idp-ca:82/affwebservices/public/saml2sso?SPID=http%3A%2F%2Ffedsrv.  
acme.com  
&ProtocolBinding=urn:oasis:names:tc:SAML:2.0:bindings:HTTP-  
Artifact
```

Optional Use of the ProtocolBinding Query Parameter

If you *do not* use the ProtocolBinding query parameter, the following conditions apply:

- If the ProtocolBinding is not specified in the unsolicited response, the profile for the Service Provider is used.
- Both profiles can be enabled for the Service Provider. If the ProtocolBinding is not in the unsolicited response, the Service Provider uses the POST profile by default.

Example: Unsolicited Response without ProtocolBinding

This link redirects the user to the Single Sign-on service. Included in this link is the Service Provider identity. The SPID query parameter indicates the identity. No ProtocolBinding query parameter exists. After the user clicks this hard-coded link, they are redirected to the local Single Sign-on service.

```
http://fedsrv.fedsite.com:82/affwebservices/public/saml2sso?SPID=
http%3A%2F%2Ffedsrv.acme.com%2Fsmidp2for90
```

RelayState

Specifies the target at the Service Provider. Use the RelayState query parameter to indicate the target destination; however, this method is optional. There can be a configuration mechanism at the Service Provider to indicate the target.

URL-encode the RelayState value.

Example

```
http://ca.sp.com:90/affwebservices/public/saml2authnrequest?ProviderID=
http%3A%2F%2Ffedsrv.acme.com%2Fsmidp2for90&
RelayState=http%3A%2F%2Fwww.spdemo.com%2Fapps%2Fapp.jsp
```

Service Provider-initiated SSO (POST or artifact binding)

A user can visit the Service Provider first and then go to an Identity Provider. Therefore, create an HTML page at the Service Provider containing hard-coded links to its AuthnRequest service. The links in the HTML page redirect the user to the Identity Provider for authentication. The links also indicate what is in the AuthnRequest.

The hard-coded link that the user selects must contain specific query parameters. These parameters are part of the HTTP GET request to the AuthnRequest service at the Service Provider.

Note: The page with these hard-coded links has to reside in an unprotected realm.

To specify the use of artifact or profile binding for the transaction, the syntax for the link is:

`http://SP_server/affwebservices/public/saml2authnrequest?ProviderID=IdP_ID&ProtocolBinding=URI_of_binding`

sp_server:port

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

IdP_ID

Specifies the identity that is assigned to the Identity Provider.

URI_for_binding

Identifies the URI of the POST or Artifact binding for the ProtocolBinding element. The SAML 2.0 specification defines this URI.

For the request to work, enable a binding for the SAML authentication scheme.

Note the following information:

- If you do not include the ProtocolBinding query parameter in the AuthnRequest, the default binding is the one defined for the authentication scheme. If you have both bindings defined in the authentication scheme, then no binding is passed in the AuthnRequest. As a result, the default binding at the Identity Provider is used.
- Artifact and POST can be enabled for the SAML authentication scheme. Include the ProtocolBinding query parameter in the link if you only want to use artifact binding.

AuthnRequest Query Parameters Used by a SiteMinder SP

A SiteMinder Service Provider can use query parameters in the links to the AuthnRequest Service. The allowable query parameters are:

ProviderID (required)

ID of the Identity Provider where the AuthnRequest Service sends the AuthnRequest message.

ProtocolBinding

Specifies the ProtocolBinding element in the AuthnRequest message. This element specifies the protocol that the Identity Provider uses to return the SAML response. If the specified Identity Provider is not configured to support the specified protocol binding, the request fails.

If you use this parameter in the AuthnRequest, you cannot include the AssertionConsumerServiceIndex parameter also. They are mutually exclusive.

Required Use of the ProtocolBinding Query Parameter

The artifact and POST binding can be enabled for an authentication scheme. If you want to use only the artifact binding, the ProtocolBinding parameter is required.

- The URI for the artifact binding from the SAML 2.0 specification is:
urn:oasis:names:tc:SAML:2.0:bindings:HTTP-Artifact
- The URI for the POST binding as from the SAML 2.0 specification is:
urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST

You do not need to set this parameter for HTTP-POST single sign-on.

Example: AuthnRequest Link with ProtocolBinding

```
http://ca.sp.com:90/affwebservices/public/saml2authnrequest?ProviderID=
http%3A%2F%2Ffedsrv.acme.com%2Fsmidp2for90&ProtocolBinding=urn:oasis:
names:tc:SAML:2.0:bindings:HTTP-Artifact
```

A user clicks the link at the Service Provider. The Federation Web Services application requests an AuthnRequest message from the local Policy Server.

Optional Use of ProtocolBinding

When you *do not* use the ProtocolBinding query parameter, the following conditions apply:

- If only one binding is enabled for the authentication scheme and the ProtocolBinding query parameter is not specified, the authentication scheme uses the enabled binding.
- If both bindings are enabled and the ProtocolBinding query parameter is not specified, POST binding is used as the default.

Note: Do not HTTP-encode the query parameters.

Example: AuthnRequest Link without ProtocolBinding

This sample link goes to the AuthnRequest service. The link specifies the Identity Provider in the ProviderID query parameter.

```
http://ca.sp.com:90/affwebservices/public/saml2authnrequest?ProviderID=
http%3A%2F%2Ffedsrv.acme.com%2Fsmidp2for90
```

A user clicks the link at the Service Provider. The Federation Web Services application requests for an AuthnRequest message from the local Policy Server.

ForceAuthn

Indicates whether the SP forces the Identity Provider to authenticate a user even if there is an existing security context for that user.

- If ForceAuthn=True in the AuthnRequest message, and a SiteMinder session exists for a particular user, the IdP rechallenges the user for credentials. If the user successfully authenticates, the IdP includes the identity information from the existing session in the assertion. The IdP discards the session that it generated for reauthentication.

Note: A user can try to reauthenticate with different credentials than the existing session. The IdP then compares the userDN and the user directory OID for the current and existing sessions. If the sessions are not for the same user, the IdP returns a SAML 2.0 response. The response indicates that the authentication has failed.

- If the SP sets ForceAuthn=True in the AuthnRequest message and there is no SiteMinder session, the SiteMinder IdP challenges the user for credentials. If the user successfully authenticates, a session is established.

Example

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

RelayState

Specifies the target at the Service Provider. You can use the RelayState query parameter to indicate the target destination, but this method is optional. Instead, you can specify the target configured in the SAML 2.0 authentication scheme. The authentication scheme also has an option to override the target with the RelayState query parameter.

URL-encode the RelayState value.

Example

```
http://www.spdemo.com:81/affwebservices/public/saml2authnrequest?ProviderID=idp.demo&RelayState=http%3A%2F%2Fwww.spdemo.com%2Fapps%2Fapp.jsp
```

IsPassive

Determines whether the Identity Provider can interact with a user. If this query parameter is set to true, the Identity Provider must not interact with the user. Additionally, the IsPassive parameter is included with the AuthnRequest sent to the Identity Provider. If this query parameter is set to false, the Identity Provider can interact with the user.

Example

```
http://www.spdemo.com:81/affwebservices/public/saml2authnrequest?ProviderID=idp.demo&RelayState=http%3A%2F%2Fwww.spdemo.com%2Fapps%2Fapp.jsp&IsPassive=true
```

AssertionConsumerServiceIndex

Specifies the index of the endpoint acting as the assertion consumer. The index tells the Identity Provider where to send the assertion response.

If you use this parameter in the AuthnRequest, you cannot include the ProtocolBinding parameter also. They are mutually exclusive.

Query Parameter Processing by a SiteMinder IdP

If a Service Provider initiates single sign-on, that Service Provider can include a ForceAuthn or IsPassive query parameter in the AuthnRequest message. When a Service Provider includes these two query parameters in the AuthnRequest message, a <stnmdr> Identity Provider handles these query parameters as follows:

ForceAuthn Handling

When a Service Provider includes ForceAuthn=True in the AuthnRequest, a SiteMinder Identity Provider takes the following actions:

- ForceAuthn=True in the AuthnRequest message, and a SiteMinder session exists for a particular user. The SiteMinder IdP rechallenges the user for credentials. If the user successfully authenticates, the IdP sends the identity information from the existing session in the assertion. The IdP discards the session that it generated for the reauthentication.

A user can try to reauthenticate with different credentials than the original session. The SiteMinder IdP compares the userDN and the user directory OID for the current and existing sessions. If the sessions are not for the same user, it returns a SAML 2.0 response. The response indicates that the authentication has failed.

- ForceAuthn=True in the AuthnRequest message and there is no SiteMinder session. The SiteMinder IdP challenges the user for credentials. If the user successfully authenticates, a session is established.

IsPassive Handling

When a Service Provider includes IsPassive in the AuthnRequest and the IdP cannot honor it, one of the following SAML responses is sent back to the Service Provider:

- IsPassive=True in the AuthnRequest message and there is no SiteMinder session. The SiteMinder Identity Provider returns a SAML response. This response includes an error message because SiteMinder requires a session.
- IsPassive=True in the AuthnRequest message and there is a SiteMinder session. The SiteMinder Identity Provider returns the assertion.
- IsPassive and ForceAuthn are in the AuthnRequest message and both are set to True. The SiteMinder Identity Provider returns an error because the request is invalid. IsPassive and ForceAuthn are mutually exclusive.

Configure Single Logout (optional)

The single logout protocol (SLO) results in the simultaneous end of all sessions for a particular user, thereby ensuring security. These session must be associated with the browser that initiated the logout. Single logout does not necessarily end all sessions for a user. For example, if the user has two browsers open, that user can establish two independent sessions. Only the session for the browser that initiates the single logout is terminated at all federated sites for that session. The session in the other browser will still be active. Single logout is triggered by a user-initiated logout.

Note: SiteMinder only supports the HTTP-Redirect binding for the single logout protocol.

By configuring the settings on the SLO tab you are informing the Identity Provider whether the Service Provider supports the single logout protocol, and if so, how single logout is handled.

If you enable single logout, you must also:

- Enable the session server at the Identity Provider using the Policy Server Management Console.
- Configure persistent sessions for the realm containing the protected resources at the Service Provider. Persistent session are configured via the FSS Administrative UI.

To configure single logout

1. Log in to the FSS Administrative UI and access the SAML Service Provider Properties dialog box for the Service Provider you want to configure.
2. From the SAML Service Provider Properties dialog box, select the SLO tab.
3. Select the HTTP-Redirect checkbox to enable single logout.

The remaining fields become active.

4. Enter values for the remaining fields, noting the following:

Validity Duration

Specifies the number of seconds that a single logout request is valid. This property is different from the Validity Duration on the SSO tab, which is for assertions. If the validity duration expires, a single logout response is generated and is sent to the entity who initiated the logout. The validity duration also depends on the skew time (set in the General tab) to calculate single logout message duration.

SLO Location URL, SLO Response Location URL, and SLO Confirm URL

Entries for these fields must start with https:// or http://.

Note: Click Help for descriptions of settings and controls, including their respective requirements and limits.

5. (Optional) Select the Reuse Session Index field to use the same session index for assertions sent to the same partner during one browser session. This option helps ensure single logout is successful with all third-party partners.

Federation Web Services redirects the user to the logout confirm page after the user's session is completely removed at the Identity Provider and all Service Provider sites.

More Information:

[Single Logout Request Validity](#) (see page 332)

Single Logout Request Validity

The SLO validity duration and Skew Time instruct the Policy Server how to calculate the total time that the single logout request is valid.

Note: The SLO Validity Duration is a different value from the SSO Validity Duration.

The two values that are relevant in calculating the logout request duration are referred to as the IssueInstant value and the NotOnOrAfter value. In the SLO response, the single logout request is valid until the NotOnOrAfter value.

When a single logout request is generated, the Policy Server takes its system time. The resulting time becomes the IssueInstant value, which is set in the request message.

The Policy Server determines when the logout request is no longer invalid. The Policy Server takes its current system time and adds the Skew Time plus the SLO Validity Duration. The resulting time becomes the NotOnOrAfter value. Times are relative to GMT.

For example, a log out request is generated at the Identity Provider at 1:00 GMT. The Skew Time is 30 seconds and the SLO Validity Duration is 60 seconds. Therefore, the request is valid between 1:00 GMT and 1:01:30 GMT. The IssueInstant value is 1:00 GMT and the single logout request message is no longer valid 90 seconds afterward.

Guidelines for the Single Logout Confirmation Page

To support single logout, have a logout confirmation page at your site. This page lets the user know they are logged out.

The logout confirmation page must satisfy the following criteria:

- If the single logout is initiated at the Service Provider, the logout confirmation page must be an unprotected local resource at the Service Provider site.
- If single logout is initiated at an Identity Provider site, the logout confirmation page must be an unprotected local resource at the Identity Provider site.
- The page cannot be a resource in a federation partner domain. For example, if the local domain is ca.com, the SLO confirmation page cannot be in the example.com domain.

To receive feedback about a logout failure, the logout confirmation page must also support the following requirements:

- Be able to handle Base 64-encoded data and read cookies.
- Contain code that looks for a SIGNOUTFAILURE cookie. This condition must be met by the logout page at the IdP and the SP. If single logout fails, the cookie is set in the browser. The cookie contains the Partner IDs of the federation sites where logout failed. These IDs are base 64-encoded. If multiple IDs are listed, they are separated by a space character.

By configuring the logout confirmation page to look for this cookie, the page can inform the user where the logout failed. This information is useful in networks where a user is logging out from multiple partner sites.

Additionally, if the SIGNOUTFAILURE cookie is found, the logout confirmation page must inform users to close the web browser to remove all session data.

Configure Identity Provider Discovery at the IdP

The Identity Provider Discovery (IPD) profile provides a common discovery service that enables a Service Provider to select a unique IdP for authentication. A prior business agreement between partners is established so that all sites in the network interact with the Identity Provider Discovery service.

This profile is useful in federated networks that have more than one partner providing assertions. A Service Provider can determine which Identity Provider it sends authentication requests for a particular user.

The IdP Discovery profile is implemented using a cookie domain that is common to the two federated partners. A cookie in the agreed upon domain contains the list of IdPs that the user has visited.

For the IDP Discovery profile, the SP has to determine the IdP to which it sends authentication requests. The user that the SP wants to authenticate must have previously visited the Identity Provider and authenticated.

At the IdP, you only enable the Identity Provider Discovery feature. No other configuration is required. Enabling the feature results in a cookie being set in the common domain at the IDP Discovery Service. This process is transparent to the user.

Enable Identity Provider Discovery Profile (optional)

For federated networks that have more than one IdP generating assertions, the Identity Provider Discovery profile enables users to select a specific IdP for authentication.

To enable the Identity Provider Discovery Profile

1. Log on to the FSS Administrative UI.
2. Open the Service Provider Properties dialog for the SP you want to modify.
3. Select the IPD tab.

The Identity Provider Discovery settings display.

4. Select the Enable checkbox.
The fields in the dialog become active.
5. Fill in the necessary fields and click OK.

Note: Set the Service URL field to the Identity Provider Discovery Profile servlet, which is:

`https://host:port/affwebservices/public/saml2ipd`

Securing the IdP Discovery Target Against Attacks

When the SiteMinder Identity Provider Discovery Service receives a request for the common domain cookie, the request includes a query parameter named `IPDTarget`. This query parameter lists a URL where the Discovery Service must redirect to after it processes the request.

For an IdP, the `IPDTarget` is the SAML 2.0 Single Sign-on service. For an SP, the target is the requesting application that wants to use the common domain cookie.

We recommend protecting the `IPDTarget` query parameter against security attacks. An unauthorized user can place any URL in this query parameter. The URL can cause a redirection to a malicious site.

To protect the query parameter against an attack, configure the Agent Configuration Object setting **ValidFedTargetDomain**. The `ValidFedTargetDomain` parameter lists all valid domains for your federated environment.

Note: The `ValidFedTargetDomain` setting is similar to the `ValidTargetDomain` setting that the Web Agent uses, but this setting is defined specifically for federation.

The IPD Service examines the `IPDTarget` query parameter. The service obtains the domain of the URL that the query parameter specifies. The IPD Service compares this domain to the list of domains specified in the `ValidFedTargetDomain` parameter. If the URL domain matches one of the configured domains in the `ValidFedTargetDomain`, the IPD Service redirects the user to the designated URL.

If there is no domain match, the IPD Service denies the user request and they receive a 403 Forbidden in the browser. Additionally, errors are reported in the FWS trace log and the `affwebservices` log. These messages indicate that the domain of the `IPDTarget` is not defined as a valid federation target domain.

If you do not configure the `ValidFedTargetDomain` setting, the service redirects the user to the target URL without performing any validation.

More information:

[Solution 7: Identity Provider Discovery Profile \(SAML 2.0\)](#) (see page 56)

Encrypt a NameID and an Assertion

You can encrypt the Name ID in an assertion or the assertion itself. Encryption adds another level of protection when transmitting the assertion.

When you configure encryption, specify the partner certificate. The certificate is in the assertion. When the assertion arrives at the Service Provider, the Service Provider decrypts the encrypted data using the associated private key.

Note: If you enable encryption, the first federation call can cause the Policy Server memory to increase to load the encryption libraries and allocate additional memory.

Enabling Encryption

To implement encryption

1. Log in to the FSS Administrative UI and access the SAML Service Provider Properties dialog box for the Service Provider you want to configure.
2. From the SAML Service Provider Properties dialog box, select the Encryption tab.

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

3. To encrypt only the Name ID, select the Encrypt Name ID checkbox.
4. To encrypt the entire assertion, select the Encrypt Assertion checkbox.

You can select the Name ID and the assertion; both can be encrypted.

5. Choose an Encryption Block Algorithm and Encryption Key Algorithm. These algorithms are defined by the WC3 XML Syntax and Processing standards.

After you select an encryption check box, the fields in the Encryption Public Key become active.

Notes:

- If you select rsa-oaep as an Encryption Key Algorithm, the minimum key size required is a 1024 bits.
- To use the aes-256 bit encryption block algorithm, install Sun's Java Cryptography Extension (JCE) Unlimited Strength Jurisdiction Policy Files. You can download these files from <http://java.sun.com/javase/downloads/index.jsp>

6. Fill-in the IssuerDN and the Serial Number fields.

The IssuerDN is the DN of the certificate issuer and its associated serial number. This information locates the certificate of the Service Provider in the key store. The data should be supplied by the Service Provider.

Additionally, the IssuerDN and Serial Number that you enter here and on the General tab must match an IssuerDN and serial number of a key stored in the Identity Provider's key store database. The key store is created using the SiteMinder keytool utility.

7. Click OK to save your changes.

Request Processing with a Proxy Server at the IdP

Before SiteMinder processes a request as an Identity Provider, it validates the message attributes using the local URL for the Federation Web Services application.

For example, an AuthnRequest message from an SP can contain the following attribute:

```
Destination="http://idp.domain.com:8080/affwebservices/public/saml2sso"
```

In this example, the destination attribute in the AuthnRequest and the address of the Federation Web Services application are the same. SiteMinder verifies that the destination attribute matches the local URL of the FWS application.

If SiteMinder sits behind a proxy server, the local and destination attribute URLs are not the same. The Destination attribute is the URL of the proxy server. For example, the AuthnRequest can include the following Destination attribute:

```
Destination="http://proxy.domain.com:9090/affwebservices/public/saml2sso"
```

The local URL for Federation Web Services, `http://idp.domain.com:8080/affwebservices/public/saml2sso`, does not match the Destination attribute so SiteMinder denies the request.

You can specify a proxy configuration to alter how SiteMinder determines the local URL for verifying a message attribute. If you specify a proxy, SiteMinder replaces the `<protocol>://<authority>` portion of the local URL with the proxy server URL. The result is a match between the two URLs.

Configure Request Processing with a Proxy Server

If your federated environment sits behind a proxy server, you must specify a proxy configuration to ensure that SiteMinder finds a match between the URL of a request's message attribute and the local proxy URL. There must be a match for the request to be processed.

When a proxy configuration is set, SiteMinder replaces the `<protocol>://<authority>` portion of the local URL with the proxy server URL, which results in a match between the two URLs.

To support federated environments that use a proxy server at the IdP

1. Log in to the FSS Administrative UI.
2. Access the SAML Service Provider Properties dialog box for the Service Provider you want to configure.

The SAML Service Provider Properties dialog opens.

3. Select the Advanced tab.
4. Enter a partial URL for the proxy server, of the form `<protocol>://<authority>` in the Server field of the Proxy group box.

For example, the proxy server configuration would be:

```
http://proxy.domain.com:9090
```

If your network includes the SPS federation gateway, the Server field must specify the SPS federation gateway host and port, for example,

```
http://sps_gateway_server.ca.com:9090
```

5. Click OK to save your changes.

The value you enter for the Server field affects the URLs for the following services at the IdP:

- Single Sign On Service
- Single Logout Service
- Artifact Resolution Service
- Attribute Service
- Authentication URL – use the proxy server URL. Once authenticated, the user is redirected to the proxy server to get to the Single Sign On Service.

The Server value becomes part of the URL used to verify SAML attributes like the Destination attribute. Essentially, if you are using a proxy server for one URL, you need to use it for all these URLs.

HTTP Error Handling at the IdP

Assertion-based single sign-on can fail at the Identity Provider for various reasons. If an HTTP error occurs, Federation Security Services provides functionality to redirect the user to different applications (URLs) for further processing. Redirection to a customized error page can take place only when the IdP has the necessary information about the Service Provider. If this information is not available when the error occurs, only the HTTP error code is returned to the browser without any redirection.

You can configure redirect URLs for HTTP handling, but they are not required.

To configure optional redirect URLs for error handling

1. From the SAML Service Provider Properties dialog, select the Advanced tab.

The Advanced tab dialog displays.

2. Click Additional Configuration.

The Additional URL Configuration dialog opens.

3. Specify a URL for one or more of the following settings:

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

- Enable Server Error URL
- Enable Invalid Request URL
- Enable Unauthorized Access URL

4. Select one of the following for the redirect mode:

- 302 No Data
- HTTP POST

5. Click OK.

Note: These redirect URLs can be used with the SiteMinder Assertion Consumer Plug-in for further assertion processing. If an assertion request fails, the plug-in can send the user to one of the redirect URLs you specify.

Customize a SAML Response Element (optional)

The Assertion Generator produces SAML assertions to authenticate users in a federated environment. You may want to modify the assertion content based on your business agreements between partners and vendors.

By configuring an Assertion Generator plug-in, you can customize the content of a SAML 2.0 response generated by the Assertion Generator.

To modify a response element using the Assertion Generator plug-in

1. Implement the plug-in class.

A sample class, `AssertionSample.java`, can be found in `sdk/samples/assertiongeneratorplugin`.

2. Configure the Assertion Generator plug-in from the Advanced tab of the SAML Service Provider Properties dialog box.

Note: Specify an Assertion Generator plug-in for each Service Provider.

- a. In the Full Java Class Name field, enter the Java class name of the plug-in. This plug-in is invoked by the Assertion Generator at run time.

The plug-in class can parse and modify the assertion, and then return the result to the Assertion Generator for final processing.

Only one plug-in is allowed for each Service Provider. For example, `com.mycompany.assertiongenerator.AssertionSample`

A sample plug-in is included in the SDK. You can view a sample assertion plug-in at `sdk/samples/assertiongeneratorplugin`.

- b. Optionally, in the Parameters field, enter the string that gets passed to the plug-in as a parameter at run time.

The string can contain any value; there is no specific syntax to follow.

Additional information about the Assertion Generator plug-in can be found as follows:

- Reference information (method signatures, parameters, return values, data types), and also the new constructor for `UserContext` class, are in the *Javadoc Reference*. Refer to the `AssertionGeneratorPlugin` interface in the Javadoc.
- Overview and conceptual information for authentication and authorization APIs is in the *SiteMinder Programming Guide for Java*.

Implement the AssertionGeneratorPlugin Interface

The first step in creating a custom assertion generator plug-in is to implement the `AssertionGeneratorPlugin` interface.

Follow these steps:

1. Provide a public default constructor method that contains no parameters.
2. Provide code so that the implementation is stateless. Many threads must be able to use a single plug-in class.

3. Implement methods in the interface to satisfy your requirements.

The implementation must include a call to the `customizeAssertion` methods. You can overwrite the existing implementations. See the following sample classes for examples:

SAML 1.x/WS-Federation

`AssertionSample.java`

SAML 2.0

`SAML2AssertionSample.java`

The sample classes are located in the directory `/sdk/samples/assertiongeneratorplugin`.

Note: The contents of the parameter string that your implementation passes into the `customizeAssertion` method is the responsibility of the custom object.

Deploy the Assertion Generator Plug-in

After you have coded your implementation class for the `AssertionGeneratorPlugin` interface, compile it and verify that SiteMinder can find your executable file.

To deploy the assertion generator plug-in

1. Compile the assertion plug-in Java file.

Compilation requires the following .jar files, which are installed with the Policy Server:

- `policy_server_home/bin/jars/SmJavaApi.jar`
- `policy_server_home/bin/thirdparty/xercesImpl.jar`
- `policy_server_home/bin/endorsed/xalan.jar`

2. In the `JVMOptions.txt` file, modify the `-Djava.class.path` value so it includes the classpath for the plug-in. This modification enables the plug-in to be loaded with the modified classpath. Locate the `JVMOptions.txt` file in the directory `installation_home\siteminder\config`.

Note: Do not modify the classpath for `xercesImpl.jar`, `xalan.jar`, or `SMJavaApi.jar`.

3. Enable the plug-in.

Enable the Assertion Generator Plug-in (SAML 2.0)

After writing an assertion generator plug-in and compiling it, enable the plug-in by configuring settings in the FSS Administrative UI. The UI parameters let SiteMinder 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 in to the FSS Administrative UI.
2. Navigate to the Service Provider Properties and access the Advanced tab.
3. Complete the following fields:

Full Java Class Name

Specify a Java class name for an existing plug-in.

Parameter

Specify a string of parameters that is passed to the plug-in specified in the Full Java Class Name field.

Note: Instead of specifying the assertion plug-in class and its parameters through the Administrative UI, you can use the Policy Management API (C or Perl) to integrate the plug-in. For more information, see the *SiteMinder Programming Guide for C* or the *SiteMinder Programming Guide for Java*.

4. Restart the Policy Server.

Restarting the Policy Server verifies that the latest version of the assertion plug-in is picked up after being recompiled.

Chapter 15: Configure SAML 2.0 Affiliations At the Identity Provider

This section contains the following topics:

[Affiliation Overview](#) (see page 343)

[Configure Affiliations](#) (see page 344)

Affiliation Overview

A SAML affiliation is a group of SAML entities that share a name identifier for a single principal.

Service Providers and Identity Providers can belong to an affiliation. However, a single entity can belong to only one affiliation. Service Providers share the Name ID definition across the affiliation. Identity Providers share the user disambiguation properties across the affiliation.

Affiliations reduce the configuration that is required at each Service Provider. Additionally, using one name ID for a principal saves storage space at the Identity Provider.

SiteMinder uses affiliations for the following functions:

- Single sign-on
- Single logout

Note: Configuring affiliations is optional.

Affiliations for Single Sign-On

In a single sign-on use case, the Service Provider sends a request for an assertion to an Identity Provider. The AuthnRequest contains an attribute that specifies an affiliation identifier.

When the Identity Provider receives the request, it takes the following actions:

- Verifies that the Service Provider is a member of the affiliation identified in the AuthnRequest.
- Generates the assertion with the Name ID that is shared by the affiliation.
- Returns this assertion to the Service Provider.

Upon receiving the assertion, authentication takes place at the Service Provider.

Affiliations for Single Logout

When a Service Provider generates a logout request, it verifies whether the Identity Provider is a member of an affiliation. The Service Provider includes an attribute in the request, which it sets to the affiliation ID. The Identity Provider receives the request and verifies that the Service Provider belongs to the affiliation identified in the attribute.

The Identity Provider obtains the affiliation Name ID from the session store of the session store. When the Identity Provider issues logout request messages to all session participants, it includes the affiliation Name ID for the members of the affiliation.

Configure Affiliations

A SAML affiliation lets you add a SAML entity to a group so it can share a name identifier for a single principal.

To configure an affiliation

1. From the menu bar, select Edit, System Configuration, Create SAML Affiliation.
The SAML Affiliation Properties dialog box opens.
2. In the top half of the dialog box, the following fields are required:
 - Name
 - Affiliation ID

Assign Name IDs to Affiliations

To assign a name ID associated with an affiliation, you need to configure the shared Name ID properties for the Service Providers belonging to the affiliation.

Note: If you use an affiliation, configuring a Name ID is required.

To configure a name ID

1. Select the Name IDs tab from the SAML Affiliation Properties dialog box.

2. Determine the value to use for the Name ID format.

The format determines the type of value used for the identifier, such as whether the format is an email address or Windows domain qualified name.

3. Choose a Name ID Type.

The type indicates if the value is static, a user attribute, or a distinguished name attribute from a user store.

If you select the DN Attribute, the Allow Nested Groups check box can also be selected. Enabling nested groups means that the user record may be a DN from a user directory record nested within another directory.

4. Depending on the Name ID Type selected, fill-in the appropriate Name ID field(s).
5. Click OK to save your changes.

Specify Users for Disambiguation for SAML Affiliations

The Users tab has no function for a site acting as an Identity Provider. Disregard this tab.

For a system acting as a Service Provider, the Users tab lets you configure the user disambiguation process.

To configure the disambiguation process for a Service Provider

1. Enter an Xpath query in the Xpath Query field that the authentication scheme uses to obtain the LoginID from the assertion.
2. Select a namespace in the Namespace list box to match the search specification to and click Edit.

The SiteMinder Authentication Scheme Namespace Mapping dialog box opens.

3. In the Search Specification field, enter the attribute that the authentication scheme uses to search a namespace, then click OK. Use %s in the entry as a LoginID variable.

Note: Click Help for descriptions of settings and controls, including their respective requirements and limits.

For example, the LoginID has a value of user1. If you specify Username=%s in the Search Specification field, the resulting string is Username=user1. This string is checked against the user store to find the correct record for authentication.

View a List of Service Providers in an Affiliation

To see a list of Service Providers that are members of the affiliation, select the SAML Service Providers tab.

This is a read-only list; you can only modify this list of affiliations from the Service Provider dialog box.

More Information:

[Configure Affiliations](#) (see page 344)

View Authentication Schemes That Use an Affiliation

To see a list of authentication schemes that use an affiliation for user disambiguation, select the SAML Auth Schemes tab.

This is a read-only list. To edit this list or the schemes themselves, you need to edit the particular scheme from the authentication scheme dialog box.

More Information:

[Configure SiteMinder as a SAML 2.0 Service Provider](#) (see page 347)

Chapter 16: Configure SiteMinder as a SAML 2.0 Service Provider

This section contains the following topics:

- [SiteMinder as a Service Provider](#) (see page 347)
- [SAML 1.x Authentication Scheme Prerequisites](#) (see page 350)
- [How to Configure a SiteMinder Service Provider](#) (see page 351)
- [Configure the SAML 2.0 Authentication Scheme](#) (see page 352)
- [Look Up User Records for SAML 2.0 Authentication](#) (see page 354)
- [Configure Single Sign-on at the SP](#) (see page 357)
- [Enable Single Logout](#) (see page 364)
- [Enforce Assertion Encryption Requirements for Single Sign-on](#) (see page 365)
- [Supply SAML Attributes as HTTP Headers](#) (see page 366)
- [IDP Discovery Configuration at the Service Provider](#) (see page 371)
- [Customize Assertion Processing with the Message Consumer Plug-in](#) (see page 373)
- [Specify Redirect URLs for Failed SAML 2.0 Authentication](#) (see page 376)
- [HTTP Error Handling for SAML 2.0 Authentication](#) (see page 378)
- [Request Processing with a Proxy Server at the SP](#) (see page 379)
- [Enable Client Certificate Authentication for the Back Channel\(optional\)](#) (see page 380)
- [How To Protect Resources with a SAML 2.0 Authentication Scheme](#) (see page 382)

SiteMinder as a Service Provider

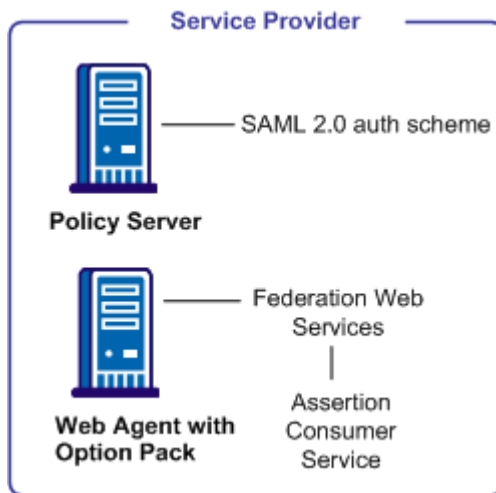
SiteMinder can consume assertions to authenticate users. If the SiteMinder sites in your federated network that have user stores, you can use SAML 2.0 authentication.

The SAML 2.0 authentication scheme lets a Service Provider in a federated network authenticate a user. It enables cross-domain single sign-on by consuming a SAML assertion from an Identity Provider, identifying a user, and establishing a SiteMinder session. After a SiteMinder session is established, the Service Provider can authorize the user for specific resources.

The following illustration shows the components for authentication at the Service Provider.

Note: A site may be both an Identity Provider and a Service Provider.

The major components for SAML 2.0 authentication are shown in the following illustration.



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

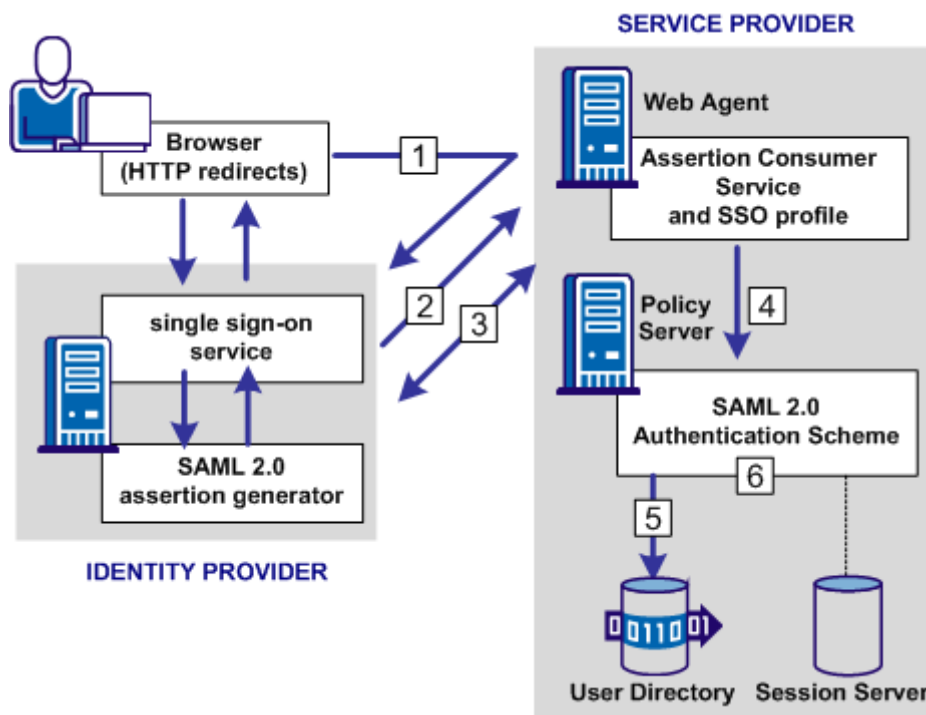
The SAML 2.0 authentication scheme is configured at the Service Provider’s Policy Server, and is invoked by the Assertion Consumer Service. This service is a component of the Federation Web Services application and is installed on the Service Provider’s Web Agent or SPS federation gateway. The Assertion Consumer Service obtains information from the SAML authentication scheme, then uses that information to extract the necessary information from a SAML assertion.

The SAML assertion becomes the user’s credentials to login to the Policy Server at the Service Provider. The user is authenticated and authorized, and if authorization is successful, the user is redirected to the target resource.

Note: The Assertion Consumer Service accepts an AuthnRequest that includes an AssertionConsumerServiceIndex value of 0. All other values for this setting will be denied.

SAML Authentication Request Process

The following illustration shows how the SAML 2.0 authentication scheme processes requests.



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

The functional flow for authentication is as follows:

1. A user makes a request for a Service Provider resource. This request goes to the AuthnRequest service at the Service Provider. The request is then redirected to the Identity Provider to obtain a SAML assertion.
2. The Identity Provider returns a response to the Service Provider.

For HTTP-POST binding, the response contains the assertion. For the HTTP-Artifact binding, the response contains a SAML artifact.

3. The Assertion Consumer Service at the Service Provider receives the response message and determines whether the POST or Artifact binding is being used.

For the HTTP-Artifact binding, the Assertion Consumer Service sends the artifact to the Identity Provider to retrieve the assertion. The Identity Provider returns a response that contains the assertion. The Assertion Consumer Service uses the response with the assertion as credentials to the Policy Server.

4. The Policy Server invokes the SAML 2.0 authentication scheme by passing the response message with the user credentials to the scheme to be authenticated.
5. The user disambiguation process begins.

6. After the disambiguation phase is complete, the SAML 2.0 authentication scheme validates the credentials in the assertion. The scheme also validates the assertion for time validity, and, if applicable, verifies that a trusted Identity Provider signed the assertion.

Note: For the POST binding, a signature is required. If a signature is not present, authentication fails. For the Artifact binding, a signed assertion is optional because the assertion is obtained over a secure channel between the Service Provider and Identity Provider.

If single logout is enabled, the SLO servlet redirects the user to a No Access URL.

More Information:

[Look Up User Records for SAML 2.0 Authentication](#) (see page 354)

SAML 1.x Authentication Scheme Prerequisites

There are several prerequisites you must fulfill before configuring a SiteMinder relying partner.

- Install the Policy Server.
For installation instructions, refer to the Policy Server Installation Guide.
- Install one of the following
 - The Web Agent and the Web Agent Option Pack. The Web Agent authenticates users and establishes a SiteMinder session. The Option Pack provides the Federation Web Services application. Be sure to deploy the FWS application on the appropriate system in your network.
 - The SPS federation gateway, which has an embedded Web Agent and has the Federation Web Services application on the embedded Tomcat web server.

For more information, see the *Web Agent Option Pack Guide* or the *Secure Proxy Server Administration Guide*.

- Set up a key database for each Policy Server that is responsible for signing, verification or both. Import private keys and certificates for functions that require verification and encrypting of messages.

The key database is a flat-file key and certificate database that lets you manage and retrieve keys and certificates required to sign and validate SAML responses used with SAML POST profile authentication.

- An asserting partner is set up within the federated network.

How to Configure a SiteMinder Service Provider

Configuring SiteMinder as a Service Provider requires the following tasks:

1. Complete the SAML 2.0 authentication scheme prerequisites.
2. Select the authentication scheme type and name it.
3. Configure disambiguation to authenticate users.
4. Configure single sign-on.

Configure a SAML authentication scheme for each Identity Provider that is a federation partner and generates assertions. Bind each scheme to a realm. The realm consists of all the URLs of the target resources requested by users. Protect these resources with a SiteMinder policy.

Note: You can also associate the scheme with a realm using a single custom authentication scheme and single realm.

Tips:

- Certain parameter values at the Identity Provider and Service Provider must match for the configuration to work. A list of those parameters is available in [Configuration Settings that Must Use the Same Values](#) (see page 471).
- Verify that you are using the correct URLs for the Federation Web Services servlets. The URLs are listed in [Federation Web Services URLs Used in SiteMinder Configuration](#) (see page 477).

Optional Tasks to Configure a Service Provider

The following configuration tasks are optional:

- Enable single logout.
- Enable encryption for Name IDs and/or assertions.
- Sign artifact resolve message and/or require signed artifact response.
- Customize assertions using the Message Consumer Plug-in.

Configure the SAML 2.0 Authentication Scheme

Before you can assign a SAML 2.0 authentication scheme to a realm, configure the scheme.

To configure the SAML 2.0 authentication scheme setup

1. Review the SAML 2.0 Authentication Scheme Prerequisites.
2. Log in to the FSS Administrative UI.
3. From the menu bar, select Edit, System Configuration, Create Authentication Scheme.

The Authentication Scheme Properties dialog opens.

4. In the Authentication Scheme Type drop-down list, select SAML 2.0 Template.

The contents of the Authentication Scheme dialog change to support the SAML 2.0 scheme.

In this dialog, you find the following:

- Scheme Common Setup section—identifies the scheme type and the protection level.
- Scheme Setup tab—identifies the Identity Provider that generates assertions for this scheme and specifies other parameters that determine how information from the assertion is processed. It also holds the D-Sig Info for digital signing.
- Advanced tab (optional)—for configuring a custom SAML 2.0 authentication scheme

Note: For HTTP-Artifact single sign-on, you can secure the artifact back channel using client certificate authentication. You can use non-FIPS 140 encrypted certificates to secure the back channel even if the Policy Server is operating in FIPS-only mode. However, for a strictly FIPS-only installation, use only certificates encrypted with FIPS 140-compatible algorithms.

5. Complete the fields.

Note: Click Help for descriptions of settings and controls, including their respective requirements and limits.

6. In the Scheme Setup tab:
 - a. Accept the value for the SAML Version field, which must be 2.0.
 - b. Configure validation of the digital signature.

By default, signature processing is enabled; the SAML 2.0 specification requires it; therefore, it *must* be enabled in a production environment. However, for debugging your initial federation setup *only*, you can temporarily disable all signature processing for the Service Provider (both signing and verification of signatures) by selecting the Disable Signature Processing option.

The value you enter for the Issuer DN field must match the issuer DN of the certificate in the smkeydatabase. We recommend that you open a command window and enter the command `smkeytool -lc` to list the certificates and view the DN to be sure that you enter a matching value.

Important! If you disable signature processing, you are disabling a mandatory security function.

- c. Select Signing Options to display the settings for [digital signing](#) (see page 353).
- d. Select the Additional Configuration button and configure at least one of the following bindings for single sign-on.
 - HTTP-Post (Additional Configuration, SSO tab)

For this binding, enter information about the certificate used to validate the signature of the posted assertion. The Issuer DN and the Serial Number together identify the certificate corresponding to the private key the IdP used to sign the assertion.
 - HTTP Redirect (Additional Configuration, SLO tab)

For this binding, enter information about the certificate used to validate the signature of the SLO request.

Digital Signing Options at the Service Provider

The SAML 2.0 authentication scheme configuration includes digital signing options for the following transactions:

- Authentication requests
- Single logout requests and responses
- Artifact resolve messages—for resolution of a SAML artifact to retrieve the assertion.
- Attribute queries—for authorizations taking place between an Attribute Authority (IdP) and a SAML Requester(SP).

To specify the signing options

1. Open the Authentication Scheme Properties dialog for the SAML 2.0 Template authentication scheme.
2. Select Signing Options.

The Signing Options dialog opens.

Complete the Signing Alias and Signature Algorithm settings.

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

Create a Custom SAML 2.0 Authentication Scheme (optional)

The Advanced tab of the Authentication Scheme Properties dialog box lets you use a custom SAML 2.0 scheme written with the SiteMinder Authentication API instead of the existing SAML 2.0 template provided by SiteMinder.

The Advanced tab contains the Library field. This field contains the name of the shared library that processes SAML artifact authentication. Do not change this value, unless you have a custom authentication scheme, written using the SiteMinder Authentication API.

The default shared library for HTML Forms authentication is `smauthhtml`.

Look Up User Records for SAML 2.0 Authentication

When you configure an authentication scheme, you define a way for the authentication scheme to look up a user in the local user store. After the correct user is located, the system generates a session for that user. Locating the user in the user store is the process of disambiguation. How Federation Security Services disambiguates a user depends on the configuration of the authentication scheme.

For successful disambiguation, the authentication scheme first determines a LoginID from the assertion. The LoginID is a SiteMinder-specific term that identifies the user. By default, the LoginID is extracted from the Name ID in the assertion. You can also obtain the LoginID using an Xpath query.

After the authentication scheme determines the LoginID, Federation Security Services checks if a search specification is configured for the authentication scheme. If no search specification is defined for the authentication scheme, the LoginID is passed to the Policy Server. The Policy Server uses the LoginID together with the user store search specification to locate the user. For example, the LoginID value is Username and the LDAP search specification is set to the uid attribute. The Policy Server searches for the user based on the uid value (`Username=uid`).

If you configure a search specification for the authentication scheme, the LoginID is not passed to the Policy Server. Instead, the search specification is used to locate a user.

You can configure user disambiguation in one of two ways:

- Locally, as part of the authentication scheme
- By selecting a configured SAML affiliation

More Information:

[Configure Disambiguation Locally as Part of the Authentication Scheme](#) (see page 355)
[Use a SAML Affiliation to Locate a User Record \(Optional\)](#) (see page 355)

Use a SAML Affiliation to Locate a User Record (Optional)

An affiliation is a group of Service Providers. Grouping Service Providers enables them to establish a link across the federated network, such that a relationship with one member of an affiliation establishes a relationship with all members of the affiliation.

All Service Providers in an affiliation share the same name identifier for a single principal. If one Identity Provider authenticates a user and assigns that user an ID, all members of the affiliation use that same name ID, reducing the configuration required at each Service Provider. Additionally, using one name ID for a principal saves storage space at the Identity Provider.

If you select an affiliation and you select to use the optional Xpath query and search specification for user disambiguation, these options are defined as part of the affiliation itself and not part of the authentication scheme.

Note: Define an affiliation before you can select it.

To select an affiliation

1. From the Authentication Scheme Properties dialog, click Additional Configuration. The SAML 2.0 Auth Scheme Properties dialog opens.
2. Select the Users tab.
3. In the SAML Affiliation drop-down field, select a predefined affiliation name. These affiliations are configured at the Identity Provider.

If you select an affiliation, the Xpath Query and Search Specification fields are disabled.

More Information:

[Configure SAML 2.0 Affiliations At the Identity Provider](#) (see page 343)

Configure Disambiguation Locally as Part of the Authentication Scheme

If you disambiguate locally, there are two steps in the process:

1. Obtain the LoginID by the default behavior or by using an Xpath query.
2. Locate the user in the user store by the default behavior or using a search specification.

Note: The use of Xpath and search specification are optional.

Obtain the LoginID

You can find the LoginID in two ways:

- Relying on the default behavior, where the LoginID is extracted from the NameID in the assertion. This option requires no configuration.
- Using an Xpath query to find the LoginID in place of the default behavior.

To use an Xpath query to determine the LoginID

1. From the Authentication Scheme Properties dialog, click Additional Configuration.
The SAML 2.0 Auth Scheme Properties dialog opens.
2. Select the Users tab.

The Users tab specifies who has access to protected resources at the Service Provider. Access to resources at the Service Provider is based on SiteMinder policies.

3. Enter an Xpath query that the authentication scheme uses to obtain a LoginID.

Note: Click Help for descriptions of settings and controls, including their respective requirements and limits.

Xpath queries must not contain namespace prefixes. The following example is an invalid Xpath query:

```
/saml:Response/saml:Assertion/saml:AuthenticationStatement/  
saml:Subject/saml:NameIdentifier/text()
```

The valid Xpath query is:

```
//Response/Assertion/AuthenticationStatement/Subject/  
NameIdentifier/text()
```

4. Click OK to save your configuration changes.

Use a Search Specification to Locate a User

After you obtain the LoginID, you can use a search specification to locate the user in place of the default behavior, where the LoginID is passed to the Policy Server.

To locate a user with a search specification

1. From the Authentication Scheme Properties dialog, click Additional Configuration.
The SAML 2.0 Auth Scheme Properties dialog opens.
2. Select the Users tab.
3. Select a namespace to match the search specification to and click Edit.
The SiteMinder Authentication Scheme Namespace Mapping dialog opens.

4. In the Search Specification field, enter a namespace attribute that the authentication scheme uses to search that namespace, then click OK. Use %s in the entry as a variable representing the LoginID.

Note: Click Help for descriptions of settings and controls, including their respective requirements and limits.

For example, the LoginID has a value of user1. If you specify Username=%s in the Search Specification field, the resulting string is Username=user1. This string is verified against the user store to find the correct record for authentication.

5. Click OK to save your configuration changes.

Configure Single Sign-on at the SP

To establish single sign-on between the Identity Provider and the Service Provider, specify the SSO bindings supported by the Service Provider.

The SSO tab configures single sign-on using the artifact or POST binding. This tab also enforces single use assertion policy for POST binding to prevent the replaying of a valid assertion.

Part of the single sign-on configuration is defining the Redirect Mode setting. The Redirect Mode specifies how Federation Security Services sends assertion attributes, if available, to the target application. You can send assertion attributes as HTTP Headers or HTTP cookies.

The HTTP headers and HTTP cookies have size restrictions that assertion attributes cannot exceed. The size restrictions are as follows:

- For HTTP headers, SiteMinder can send an attribute in a header up to the web server size limit for a header. Only one assertion attribute per header is allowed. See the documentation for your web server to determine the header size limit.
- For HTTP cookies, SiteMinder can send a cookie up to the size limit for a cookie. Each assertion attribute is sent as its own cookie. The cookie size limit is browser-specific, and that limit is for all attributes being passed to the application, not for each attribute. See the documentation for your web browser to determine the cookie size limit.

To configure single sign-on

1. From the Authentication Scheme Properties dialog, click Additional Configuration. The SAML 2.0 Auth Scheme Properties dialog opens.
2. Select the SSO tab.

3. Complete entries for the fields on the SSO tab.

The following are required fields:

- Redirect Mode
- SSO Service
- Audience
- HTTP-Artifact or HTTP-Post

If you select HTTP-Artifact as the binding, fill in the Resolution Service, Authentication, SP Name, and Password fields.

4. Specify a target resource for single sign-on to work. The target specifies the requested resource at the destination Service Provider site and it is required.
5. In the Bindings section, you can select both HTTP-Artifact and HTTP-Post.

If HTTP-POST is selected and artifact is not selected, only the POST binding is accepted from the Identity Provider. If no binding is specified, the default is HTTP-artifact.

If you select HTTP-Artifact binding,

- Enable the session server to store the assertion before it is retrieved. Instructions are located in [Storing User Session, Assertion, and Expiry Data](#) (see page 175).
- Configure the backchannel to select the type of authentication scheme protecting the Artifact Resolution Service, which retrieves the assertion at the Identity Provider. Instructions are located in [Configure the Backchannel for HTTP Artifact SSO](#) (see page 359).
- Optionally, add an entry for the Index field for each binding.

If you have multiple endpoints, you can configure indexed endpoints. The entry you include here is included by the Service Provider as a query parameter in the AuthnRequest that gets sent to the single sign-on service at the Identity Provider.

6. The following are other optional features you can select:

- Enhanced Client and Proxy Profile
- Sign AuthnRequests checkbox--tells the Policy Server at the Service Provider to sign the AuthnRequest after it is generated. This check box is required if the Identity Provider requires signed AuthnRequests. The AuthnRequest Service redirects the signed AuthnRequest to the Single Sign-on Service URL.
- Allow IdP to Create New User Identifier

More Information:

[Configure the Authentication Scheme that Protects the Artifact Service](#) (see page 236)

Configure the Backchannel for HTTP-Artifact SSO

If you select the HTTP-Artifact binding for single sign-on, configure the authentication scheme for the back channel that communicates with the Artifact Resolution Service. This service retrieves the assertion from the Identity Provider.

To configure the backchannel

1. If necessary, log on to the FSS Administrative UI.
2. Navigate to the Authentication Scheme Properties dialog.
3. Click Additional Configuration.

The SAML 2.0 Auth Scheme Properties dialog opens.

4. Select the Backchannel tab.
5. Complete all the fields on the dialog.

Important! If you are using basic authentication for the backchannel authentication scheme, the value of the SP Name field is the name of the Service Provider. No additional configuration is necessary. If you are using client certificate authentication for the backchannel, the value of the SP Name field must be the alias of the client certificate stored in the smkeydatabase. The SP uses the certificate as a credential to gain access to the Artifact Resolution Service.

6. Click OK to save your configuration.

More Information:

[WebLogic Configuration Required for Back Channel Authentication](#) (see page 293)

Enforcing a Single Use Policy to Enhance Security

A single use policy prevents SAML 2.0 assertions from being reused at a Service Provider to establish a second session. This feature applies to assertions that arrive by way of the POST binding.

Note: Single use policy feature is enabled by default when you select the HTTP-POST binding.

Designating an assertion for one time use is an additional security measure for authenticating across a single sign-on environment. From a browser, an attacker can acquire a SAML assertion that has been used to establish a SiteMinder session. The attacker can then POST the assertion to the Assertion Consumer Service at the Service Provider to establish a second session. However, if the assertion is designated for one-time use, this type of risk is mitigated.

SiteMinder enforces a single use policy using expiry data. Expiry data is time-based data about the assertion. The SAML 2.0 authentication scheme stores the expiry data in the session store. Expiry data verifies that a SAML 2.0 POST assertion is only used a single time.

How the Single Use Policy is Enforced

Upon successful validation of a SAML 2.0 assertion, the authentication scheme writes assertion data in the expiry data table. The data includes an assertion ID key and an expiration time. The session store management thread in the Policy Server deletes expired data from the expiry data table.

If the scheme tries to validate assertion data and an expiry data entry has the same assertion ID key, writing assertion data fails. If the scheme cannot write to the expiry table, the SAML 2.0 authentication scheme denies the authentication in the same manner as an invalid assertion.

If the database is unavailable, single use of the assertion cannot be enforced. Consequently, the authentication scheme denies the request and the assertion is not reused.

Configure a Single Use Policy

To configure a single use policy

1. From the Authentication Scheme Properties dialog box, click Additional Configuration.

The SAML 2.0 Auth Scheme Properties dialog box opens.

2. Select the SSO tab.
3. Select the HTTP-Post.

The Enforce Single Use Policy checkbox will also be selected by default.

4. Enable the session server.

More Information:

[Enforcing a Single Use Policy to Enhance Security](#) (see page 359)

[Storing User Session, Assertion, and Expiry Data](#) (see page 175)

Permit the Creation of a Name Identifier for SSO

As part of a single sign-on request, a Service Provider can generate an AuthnRequest that includes an attribute named AllowCreate, which is set to true. The Service Provider wants to obtain an identity for the user. Upon receiving the AuthnRequest, the Identity Provider generates an assertion. The Identity Provider searches the appropriate user record for the assertion attribute serving as the Name ID. If the Identity Provider cannot find a value for the NameID attribute, it generates a persistent identifier. The Allow/Create feature enables the creation of the identifier.

The persistent identifier is a randomly generated ID. The Identity Provider uses this identifier as the value of the Name ID attribute and places it in the assertion. The Identity Provider then returns the assertion to the Service Provider. For example, if the NameID attribute is set to telephone and there is no value for telephone in the user record, the NameID is set to the randomly generated identifier.

When the Service Provider receives the assertion, the SAML 2.0 authentication scheme processes the response. The scheme then performs a user lookup in its local user store. If the Service Provider locates the user record, it grants the user access.

Enable the Allow/Create feature at the Identity Provider for the Identity Provider to generate a unique identifier. The Identity Provider only generates the identifier if the feature is enabled. The normal flow of assertion generation continues after an entry is made in the Identity Provider log file that a unique identifier was not created.

Include an Allow/Create Attribute in Authentication Requests

To include the Allow/Create attribute in an AuthnRequest message

1. Log in to the Policy Server.
2. Access the appropriate SAML 2.0 authentication scheme or create a new scheme.
3. From the Authentication Scheme Properties dialog box, click Additional Configuration.

The SAML 2.0 Auth Scheme Properties dialog box opens.

4. Select the SSO tab.
5. Check the Allow IDP to Create New Identifier check box.
6. Click OK.

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.

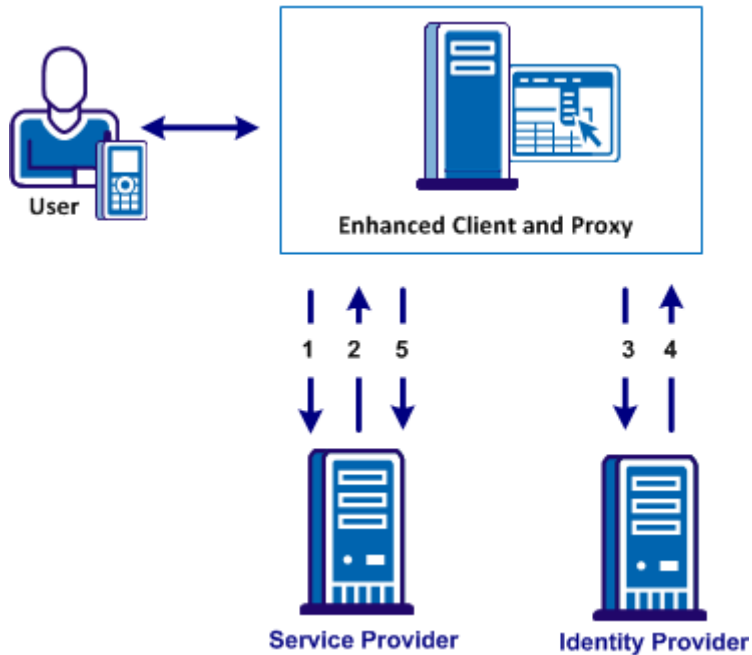
The ECP profile enables single sign-on when the Identity Provider and Service Provider cannot communicate directly. The ECP acts as the intermediary between the Service Provider and the Identity Provider.

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. SiteMinder 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 Service Provider

To configure ECP with SiteMinder, enable the feature at the Identity Provider and the Service Provider. The following procedure is for a SiteMinder Service Provider.

Follow these steps:

1. When a user requests a protected resource at the Service Provider, direct the request 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 FSS Administrative UI at the Service Provider.
3. Navigate to the SSO tab for the authentication scheme you want to modify.
4. Fill out the required single sign-on fields to configure single sign-on.
5. Select the Enhanced Client and Proxy Profile check box.
6. Click OK.

The SiteMinder Service Provider can now process ECP calls.

Note: A single SAML Service Provider object can handle artifact, POST, SOAP, and PAOS bindings for single sign-on requests. SOAP and PAOS are the bindings for the ECP profile. The Identity Provider and Service Provider determine the binding being used based on the parameters in a request.

Enable Single Logout

The single logout (SLO) profile allows near-simultaneous logout of all sessions that a specific session authority provides and which are associated with a particular user. The user initiates the logout directly. A session authority is the authenticating entity that has initially authenticated the user. In most cases, the session authority is the Identity Provider.

Single logout helps ensure that no sessions are left open for unauthorized users to gain access to resources at the Service Provider.

The user can initiate single logout service from a browser by clicking a link at the Service Provider or at the Identity Provider. The user clicks the logout link which points to an SLO servlet. This servlet, which is a component of Federation Web Services, processes logout requests and responses coming from a Service Provider or Identity Provider. The servlet does not need to know the originator of the request or response. The servlet uses the SiteMinder session cookie to determine the session to log out.

More information

[Configure Single Logout \(optional\)](#) (see page 331)

Bindings for Single Logout

The single logout feature transports messages using the HTTP-Redirect binding. This binding determines how SAML protocol messages are transported using HTTP redirect messages, which are 302 status code responses.

Configure Single Logout

To configure single logout

1. From the Authentication Scheme Properties dialog box, click Additional Configuration.
The SAML 2.0 Auth Scheme Properties dialog box opens.
2. Select the SLO tab.
3. Select the HTTP-Redirect checkbox.
The rest of the fields become active.
4. Fill in the remaining fields. Validity Duration and SLO Location URL are the two required fields.
5. Enable the session server.

More Information:

[Storing User Session, Assertion, and Expiry Data](#) (see page 175)

Enforce Assertion Encryption Requirements for Single Sign-on

The encryption feature specifies that the authentication scheme processes only an encrypted assertion or Name ID in the assertion.

For added security, the Identity Provider can encrypt the Name ID, user attributes, or the entire assertion. Encryption adds another level of protection when transmitting the assertion. When encryption is enabled at the Identity Provider, the certificate (public key) is used to encrypt the data. When the assertion arrives at the Service Provider, it decrypts the encrypted data with the associated private key.

When you configure encryption at the Session Provider, the assertion must contain an encrypted Name ID or assertion or the Service Provider rejects the assertion.

Set Up Encryption for SSO

To enforce encryption requirements

1. From the Authentication Scheme Properties dialog box, click Additional Configuration.
The SAML 2.0 Auth Scheme Properties dialog box opens.
2. Select the Encryption tab.
3. To require that only the Name ID be encrypted, select the Require Encrypted Name ID checkbox.
4. To require that the entire assertion be encrypted, select the Require Encrypted Assertion checkbox.
You can select the Name ID and the assertion.
5. Optionally, specify an alias for the private key that will be used to decrypt any encrypted data in the assertion received from the Identity Provider.
6. Click OK to save your changes.

Note: If you do not select the Encrypted Name ID or the Encrypted Assertion check box, the Service Provider accepts encrypted and clear-text Name IDs and assertions.

Supply SAML Attributes as HTTP Headers

An assertion response can include attributes in the assertion. These attributes can be supplied as HTTP header variables so a client application can use them for finer grained access control.

The benefits of including attributes in HTTP headers are as follows:

- HTTP headers are not persistent. They are present only within the request or response that contains them.
- HTTP headers, as supplied by the SiteMinder Web Agent, are not visible in the browser, which reduces security concerns.

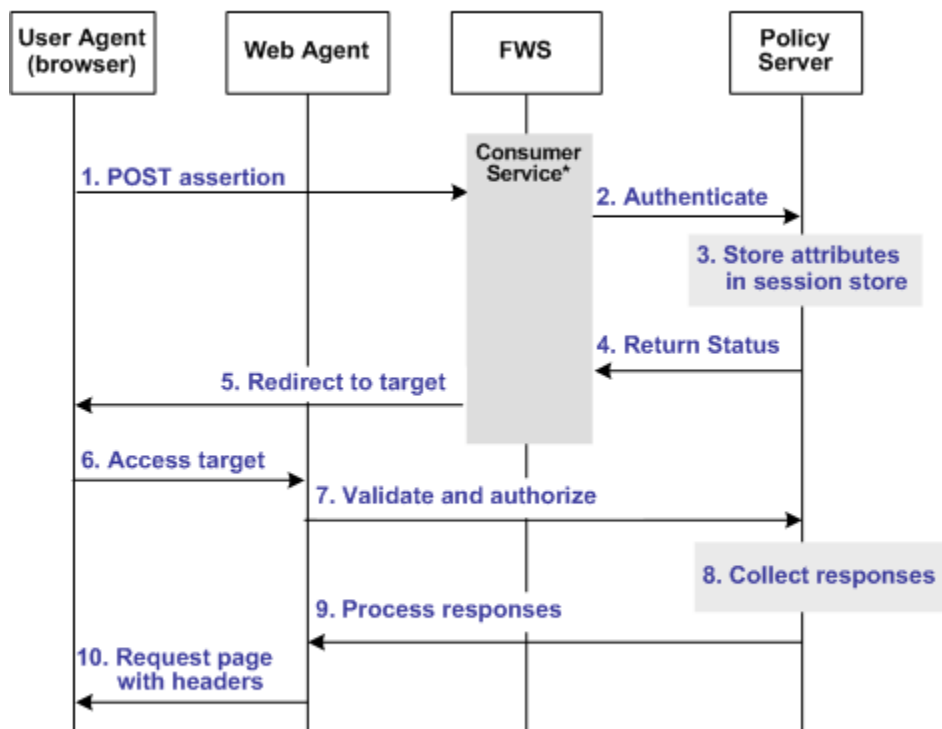
Note: The HTTP headers have size restrictions that the attributes cannot exceed. SiteMinder can send an attribute in a header up to the web server size limit for a header. Only one assertion attribute per header is allowed. See the documentation for your web server to determine the header size limit.

Use Case for SAML Attributes As HTTP Headers

During authentication, a series of SAML attributes are extracted from an assertion and supplied as HTTP headers. During the authorization process, these headers are returned to the customer application.

The following flow diagram shows the sequence of events at runtime:

Processing Headers as Attributes at the Consumer



*Consumer service can be one of the following:
 –SAML Credential Collector (SAML 1.x)
 –Assertion Consumer Service (SAML 2.0)
 –Security Token Consumer Service (WS-Federation)

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

To process the attributes as HTTP headers, the sequence of events is as follows:

1. After the assertion is generated at the asserting party, it sends the assertion to the appropriate consumer service at the relying party. The delivery mechanism (POST or Artifact or WS-Fed) is irrelevant.

Note: The consumer service can be the SAML credential collector (SAML 1.x), the Assertion Consumer Service (SAML 2.0), or Security Token Consumer Service (WS-Federation).

2. The consumer service calls its local Policy Server to use the configured authentication scheme to authenticate the user with the assertion.

3. If the authentication scheme redirect mode parameter is set to PersistAttributes, the Policy Server caches the attributes in the session store as session variables.
4. The result of the authentication is returned to the consumer service.
5. The consumer service redirects the browser to the protected target resource.
6. The browser tries to access the target resource.
7. The Web Agent calls the Policy Server to validate the user session and to verify that the user is authorized to access the target resource.
8. The Policy Server retrieves the attributes by a configured response.
9. The Policy Server processes the responses and sends the attributes to the Web Agent.
10. The Web Agent sets the HTTP headers as necessary.

Configuration Overview to Supply Attributes as HTTP Headers

Several configuration steps are required to retrieve the SAML attributes cached in the session store and provide them as HTTP headers.

Follow these steps:

1. Select PersistAttributes as the redirect mode for the SAML authentication scheme, which enables the SAML Attributes to be returned as HTTP headers.
2. Configure an authorization rule for the realm that contains the target resource.
3. Set PersistentRealm in the realm protecting the target resource.
4. Configure a response that uses the active response type for each SAML attribute to be supplied as a header.
5. Create a policy that binds the authorization rule and active response to implement the user of attributes as HTTP headers.

Set the Redirect Mode to Store SAML Attributes

After the consuming partner authenticates the user with the SAML assertion, the SAML attributes are written to the session store. The browser is then redirected to the target resource.

To redirect the browser with the attribute data

1. Log in to the FSS Administrative UI.
2. Access the SAML authentication scheme properties dialog.
The properties dialog opens.

3. Set the Redirect Mode parameter to PersistAttributes.

For SAML 1.x, the Redirect Mode is on the Scheme Setup tab. For SAML 2.0 and WS-Federation, the Redirect Mode is on the SSO tab accessed from the authentication scheme properties dialog.

4. Click OK to save your changes.

The redirect mode is now set to pass on the attribute data.

Create an Authorization Rule to Validate Users

For the realm containing the protected target resource, create a rule that is triggered during the authorization process to retrieve the SAML attributes from the session store.

The rule is based on an authorization event (`onAccessAccept`). Because the user has already been authenticated by the FWS application, the Web Agent cannot reauthenticate the user and pass on the HTTP headers. The retrieval of the attributes happen during the authorization stage.

To create an `OnAccessAccept` Rule for the realm

1. Log on to the FSS Administrative UI.
2. From the Domains tab, navigate to the realm which protects the target resource.
3. Select the realm with the target resource and select Create Rule under Realm.
The Rule Properties dialog opens.
4. Enter a name in the Name field that describes the rules purpose as an authorization rule.
5. Select the realm protecting the target resource for the Realm field.
6. Enter an asterisk (*) in the Resource field.
7. Select Authorization events and `OnAccessAccept` in the Action section.
8. Verify that Enabled is selected in the Allow/Deny and Enable/Disable section.
9. Click OK to save the rule.

The authorization rule is now defined for the realm with the protected resource.

Configure a Response to Send Attributes as HTTP Headers

Configure a response that sends the SAML attributes as HTTP headers to the Web Agent. The Web Agent processes the response and makes the header variables available to the client application.

To create a response to send the attributes as headers

1. Log on to the FSS Administrative UI.
2. From the Domains tab, select the domain that contains the target resource.
3. In the domain, select the Responses object and create a response.
The Response Properties dialog opens.
4. Click Create.
The Response Attribute dialog opens.
5. Select WebAgent-HTTP-Header-Variable in the Attribute field.
6. Select Active Response in the Attribute Kind section.
7. Complete the fields in the Attribute Fields section as follows:

Variable Name

Specify the name you want for the header variable. You assign this name.

Library Name

smfedattrresponse

This value must be the entry for this field.

Function Name

getAttributeValue

This value must be the entry for this field.

Parameters

Specify the name of the attribute as it appears in the assertion.

An agreement between you and your federated partner determines the attributes that are in the assertion.

8. Click on OK to save the attribute.
9. Repeat the procedure for each attribute that must become an HTTP header variable. You can configure many attributes for a single response.

The response sends the attributes on to the Web Agent to become HTTP headers.

Create a Policy to Implement Attributes as HTTP Headers

To implement the use of SAML attributes as HTTP headers, you must group together the authorization event rule and active response in a policy.

To create the policy to generate HTTP Headers from SAML attributes

1. Log on to the FSS Administrative UI.
2. From the Domains tab, select the domain that contains the target resource.
3. In the domain, select the Policies object and create a policy.
The Policy Properties dialog opens.
4. Enter a descriptive name in the Name field.
5. Select the users that must have access to the protected resource in the Users tab.
6. Add the authorization rule you created previously on the Rules tab.
7. Select the authorization rule and click Set Response.
The Available Responses dialog opens.
8. Select the active response you created previously and click OK.
You return to the Rules tab. The response appears with the authentication rule.
9. Click OK to save the policy.

The policy that enables SAML attributes to be used as HTTP headers is complete.

IDP Discovery Configuration at the Service Provider

The Identity Provider Discovery (IPD) profile provides a common discovery service that enables a Service Provider to select a unique IdP for authentication. A prior business agreement between partners is established so that all sites in the network interact with the Identity Provider Discovery service.

This profile is useful in federated networks that have more than one partner providing assertions. A Service Provider can determine which Identity Provider it sends authentication requests for a particular user.

The IdP Discovery profile is implemented using a cookie domain that is common to the two federated partners. A cookie in the agreed upon domain contains the list of IdPs that the user has visited. The SP has to redirect the user to the IdP Discovery Service to retrieve the common domain cookie. The cookie contains the list of IdPs that the user has already visited. From this list, the SP chooses the correct Identity Provider and then sends an AuthnRequest.

Note: The user requiring authentication must have previously visited the Identity Provider and authenticated.

IdP Discovery occurs as follows:

1. The browser requests the site selection page at the SP.
This site selection page is aware of the IdP Discovery Service URL.
2. The site selection page redirects the user to the IdP Discovery Service URL in the common domain. The redirect URL contains a query parameter indicating that it wants the Common Domain Cookie.
3. The IdP Discovery Service retrieves the value of the Common Domain Cookie and sets it as a query parameter. The service then redirects the user back to the site selection page at the SP.
4. The SP populates the site selection page with IdP IDs, which are URIs at which the user has previously authenticated.
5. The user selects an IdP to perform the user authentication.

More information:

[Configure Identity Provider Discovery at the IdP](#) (see page 334)

Securing the IdP Discovery Target Against Attacks

When the SiteMinder Identity Provider Discovery Service receives a request for the common domain cookie, the request includes a query parameter named IPDTarget. This query parameter lists a URL where the Discovery Service must redirect to after it processes the request.

For an IdP, the IPDTarget is the SAML 2.0 Single Sign-on service. For an SP, the target is the requesting application that wants to use the common domain cookie.

We recommend protecting the IPDTarget query parameter against security attacks. An unauthorized user can place any URL in this query parameter. The URL can cause a redirection to a malicious site.

To protect the query parameter against an attack, configure the Agent Configuration Object setting **ValidFedTargetDomain**. The ValidFedTargetDomain parameter lists all valid domains for your federated environment.

Note: The ValidFedTargetDomain setting is similar to the ValidTargetDomain setting that the Web Agent uses, but this setting is defined specifically for federation.

The IPD Service examines the IPDTarget query parameter. The service obtains the domain of the URL that the query parameter specifies. The IPD Service compares this domain to the list of domains specified in the ValidFedTargetDomain parameter. If the URL domain matches one of the configured domains in the ValidFedTargetDomain, the IPD Service redirects the user to the designated URL.

If there is no domain match, the IPD Service denies the user request and they receive a 403 Forbidden in the browser. Additionally, errors are reported in the FWS trace log and the affwebservices log. These messages indicate that the domain of the IPDTarget is not defined as a valid federation target domain.

If you do not configure the ValidFedTargetDomain setting, the service redirects the user to the target URL without performing any validation.

More information:

[Solution 7: Identity Provider Discovery Profile \(SAML 2.0\)](#) (see page 56)

Customize Assertion Processing with the Message Consumer Plug-in

The message consumer plug-in is a Java program that implements the Message Consumer Plug-in. The plug-in lets you implement your own business logic for processing assertions, such as rejecting an assertion and returning a status code. This additional processing works together with the standard processing of an assertion.

Note: For more information about status codes for authentication and disambiguation, see the *SiteMinder Programming Guide for Java*.

During authentication, SiteMinder first tries to process the assertion by mapping a user to its local user store. If SiteMinder cannot find the user, it calls the postDisambiguateUser method of the message consumer plug-in.

If the plug-in successfully finds the user, SiteMinder proceeds to the second phase of authentication. If the plug-in cannot map the user to a local user store, the plug-in returns a UserNotFound error. The plug-in can optionally use the redirect URL feature. Without the consumer plug-in, the redirect URLs are based on the error that the SAML authentication scheme generates.

During the second phase of authentication, SiteMinder calls the `postAuthenticateUser` method of the message consumer plug-in, if the plug-in is configured. If the method succeeds, SiteMinder redirects the user to the requested resource. If the method fails, you can configure the plug-in to send the user to a failure page. The failure page can be one of the redirect URLs that you can specify with the authentication scheme configuration.

Additional information about the message consumer plug-in can be found as follows:

- Reference information (method signatures, parameters, return values, data types), and the constructor for `UserContext` class, are in the *Java Developer Reference*. Refer to the `MessageConsumerPlugin` interface.
- Overview and conceptual information about authentication and authorization APIs, see the *SiteMinder Programming Guide for Java*.

To configure the plugin

1. Install the SiteMinder SDK, if you have not done so already.
2. Implement the `MessageConsumerPlugin.java` interface, which is part of the SiteMinder SDK.
3. Deploy your message consumer plug-in implementation class.
4. Enable the message consumer plug-in in the Administrative UI.

Implement the MessageConsumerPlugin Interface

Create a custom message consumer plug-in by implementing the `MessageConsumerPlugin.java` interface. The minimum requirements for the implementation class are listed in the following procedure.

Follow these steps:

1. Provide a public default constructor method that contains no parameters.
2. Provide code so that the implementation is stateless. Many threads must be able to use a single plug-in class.
3. Implement methods in the interface as your requirements demand.

The `MessageConsumerPlugin` includes the following four methods:

init()

Performs any initialization procedures that the plug-in requires. 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. SiteMinder calls this method once for each plug-in instance, when SiteMinder is shutting down.

postDisambiguateUser()

Provides processing to disambiguate a user when the authentication scheme is unable to do so. Alternatively, this method can add data for new federation users to a user store. This method receives the decrypted assertion. The decrypted assertion is added to the properties map passed to plug-in under the key "_DecryptedAssertion".

postAuthenticateUser()

Provides additional code to determine the final outcome of assertion processing, regardless of whether the Policy Server processing is a success or failure.

SiteMinder provides the following samples of the Message Consumer plug-in class:

MessageConsumerPluginSample.java in
installation_home\sdk\samples\messageconsumerplugin

MessageConsumerSAML20.java in
installation_home\sdk\samples\authextensionsaml20

Deploy a Message Consumer Plug-in

After you have coded your implementation class for the MessageConsumerPlugin interface, compile it and verify that SiteMinder can find your executable file.

To deploy the Message Consumer Plugin:

1. Compile the MessageConsumerPlugin Java file. The file requires the following dependent libraries, which are installed with the Policy Server:

installation_home\siteminder\bin\jars\SmJavaApi.jar

An identical copy of SmJavaApi.jar is installed with SiteMinder SDK. The file is in the directory *installation_home*\sdk\java\SmJavaApi.jar.

You can use either of them at development time.

2. When a plug-in class is available, in a folder or a jar file, modify the -Djava.class.path value in the JVMOptions.txt file. This step enables the plug-in class to load with the modified classpath. Locate the JVMOptions.txt file in the directory *installation_home*\siteminder\config.

Note: Do not modify the classpath for the existing xerces.jar, xalan.jar, or SmJavaApi.jar.

3. Restart the Policy Server to pick up the latest version of MessageConsumerPlugin. This step is necessary each time the plug-in Java file is recompiled.
4. Enable the plug-in.

Enable the Message Consumer Plug-in (SAML 2.0)

To configure the message consumer plug-in

1. Log in to the FSS Administrative UI
2. Navigate to the Authentication Scheme Properties dialog box.
3. Click Additional Configuration.

The SAML 2.0 Auth Scheme Properties dialog box opens.

4. Select the Advanced tab.
5. Complete the following fields:

Full Java Class Name

Specify the Java class name for the plug-in, For example, a sample class included with the SiteMinder SDK is:

```
com.ca.messageconsumerplugin.MessageConsumerPluginSample
```

Parameter

Specify a string of parameters to be passed to the plug-in specified in the Full Java Class Name field.

Note: Specify a Message Consumer plug-in for each authentication scheme.

6. Click OK to save the changes.
You return to the Authentication Scheme Properties dialog.
7. Click OK to exit the dialog.
8. Restart the Policy Server.

Specify Redirect URLs for Failed SAML 2.0 Authentication

If a Service Provider cannot authenticate a user during a single sign-on transaction, that user can be redirected to a customized URL for further processing.

You can configure several optional redirect URLs for failed authentication. The redirect URLs allow finer control over where a user is redirected if the assertion is not valid. For example, if a user cannot be located in a user store, you can fill in a User Not Found redirect URL and send the user to a registration page.

You can configure the following:

- Status redirect URLs
- HTTP error redirect URLs

Note: Configuring redirect URLs is not required.

The Status Redirect URLs on the Advanced tab are redirect URLs for specific status conditions. These conditions include a user is not found, the single sign-on message is invalid, or the user credentials are not accepted. If any of the conditions occur, redirect URLs can send the user to an application or a customized error page for further action.

The Additional URL Configuration dialog is where you configure redirect URLs to handle HTTP 500, 400, 405, and 403 error conditions. If any of these errors occur, redirect URLs can send the user to an application or a customized error page for further action.

Redirection to these customized URLs can take place only when enough information about the Identity Provider is provided to the Service Provider. For example, if during a request there is an issue in retrieving certificate information from smkeydatabase, the user is redirected to Server Error URL specified. However, if a request contains an invalid IdP ID, no redirection happens and the HTTP error code 400 is returned to the browser.

To configure optional redirect URLs for failed authentication

1. Select the SAML 2.0 authentication scheme you want to modify.
2. Select Additional Configuration on the Scheme Setup tab.

The SAML 2.0 Auth. Scheme Properties dialog opens.

3. Select the Advanced tab.
4. Fill in a URL for one or more of the following fields:
 - Redirect URL for the User Not Found status
 - Redirect URL for the Invalid SSO Message status
 - Redirect URL for the Unaccepted User Credential (SSO Message) status

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

Federation Web Services handles the errors by mapping the authentication reason into one of the configured redirect URLs, then the user can be redirected to that URL to report the error.

5. Select one of the following redirect modes:
 - 302 No Data
 - HTTP POST
6. Do one of the following:
 - Click OK to save your changes.
 - Select Additional URL Configuration for more redirect URLs that you can configure.

7. If you select Additional URL Configuration, specify a URL for one or more of the following fields:
 - Enable Server Error URL
 - Enable Invalid Request URL
 - Enable Unauthorized Access URL
8. Select one of the following redirect modes:
 - 302 No Data
 - HTTP POST
9. Click OK to save your changes.

Note: These redirect URLs can be used with the SiteMinder Message Consumer Plug-in for further assertion processing. If authentication fails, the plug-in can send the user to one of the redirect URLs you specify.

HTTP Error Handling for SAML 2.0 Authentication

you can configure several optional error redirect URLs if an http error occurs during the authentication process at the Service Provider.

Redirecting a user to a customized error page can take place only when there is sufficient information about the Identity Provider so that the browser can locate the error page. If this information is not available when the error occurs, the HTTP error code is returned to the browser without any redirection.

You can configure redirect URLs for HTTP handling, but they are not required.

To configure redirect URLs for error handling

1. From the SAML 2.0 Auth Scheme Properties dialog, select the Advanced tab.
2. Select Additional URL Configuration.
The Additional URL Configuration dialog opens.
3. Select one or more of the following error URL settings:
 - Enable Server Error URL
 - Enable Invalid Request URL
 - Enable Unauthorized Access URL

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

4. Specify a URL for each setting you selected.
5. Select one of the following redirect modes for each URL you enabled:
 - 302 No Data
 - HTTP POST.
6. Click OK.

Request Processing with a Proxy Server at the SP

When SiteMinder receives certain requests at the SP, it validates the message attributes. SiteMinder verifies the attributes using the local URL for Federation Web Services application. After verification, SiteMinder processes the request.

For example, a logout request message can contain the following attribute:

```
Destination="http://sp.domain.com:8080/affwebservices/public/saml2slo"
```

In this example, the destination attribute in the logout message and the address of the Federation Web Services application are the same. SiteMinder verifies that the destination attribute matches the local URL of the FWS application.

If the SiteMinder sits behind a proxy server, the local and destination attribute URLs are not the same. The destination attribute is the URL of the proxy server. For example, the logout message can include the following destination attribute:

```
Destination="http://proxy.domain.com:9090/affwebservices/public/saml2slo"
```

The local URL for Federation Web Services, `http://sp.domain.com:8080/affwebservices/public/saml2slo`, does not match the Destination attribute so the request is denied.

You can specify a proxy configuration to alter how SiteMinder determines the local URL used for verifying the message attribute of a request. In a proxy configuration, SiteMinder replaces the `<protocol>://<authority>` portion of the local URL with the proxy server URL. This replacement results in a match between the two URLs.

Configure Request Processing with a Proxy Server at the SP

To support federated environments that use a proxy server at the SP

1. Log in to the FSS Administrative UI.
2. Access the SAML 2.0 Auth Scheme Properties dialog.

The SAML 2.0 Auth Scheme Properties dialog opens.
3. Select the Advanced tab.

4. Enter a partial URL for the proxy server, in the format `<protocol>://<authority>` in the Server field of the Proxy section.

For example, the proxy server configuration would be:

`http://proxy.domain.com:9090`

If your network includes the SPS federation gateway, the Server field must specify the SPS federation gateway host and port, for example,

`http://sps_federation_gateway.domain.com:9090`

5. Click OK to save your changes.

The Server configuration affects the URLs for the following services at the SP:

- Assertion consumer Service
- Single Logout Service

The Server value becomes part of the URL used to verify SAML attributes like the Destination attribute. Essentially, if you are using a proxy server for one URL, you need to use it for all these URLs.

Enable Client Certificate Authentication for the Back Channel(optional)

If you have configured single sign-on with the artifact profile, you can select client certificate authentication to protect the Assertion Retrieval Service at the producer. This service retrieves the assertion and sends it to the consumer.

Note: Client certificate authentication is optional; you can also use Basic authentication.

The SAML credential collector invokes the SAML artifact authentication scheme. The SAML credential collector collects information from the scheme to retrieve the SAML assertion from the Producer. You are required to specify the authentication method for the realm that contains the Assertion Retrieval Service. The SAML credential collector determines what type of credentials to provide to retrieve the assertion.

If the Assertion Retrieval Service is part of a realm using a client certificate authentication scheme, complete these configuration tasks:

- At the Consumer, select the client certificate option to indicate that a certificate provides credentials.
- At the Producer, create a policy to protect the Assertion Retrieval Service.

The process of enabling client certificate authentication includes the following:

1. Add a client certificate to the certificate data store.
2. Select the client certificate option for back channel authentication.

Configure the Client Certificate Authentication at the Relying Party

The process of enabling client certificate authentication includes the following:

1. Select the Client Cert Option for Authentication
2. Add a Client Certificate to the Smkeydatabase

Select the Client Cert Option for Authentication

For the consumer to present a certificate as credentials when trying to access the Assertion Retrieval Service at the producer, select the client certificate option.

To select the client certificate option:

1. Go to the Scheme Setup tab of the SAML Artifact Authentication scheme dialog box.
2. Select Client Cert for the Authentication field.

Add a Client Certificate to smkeydatabase

When you are adding a client certificate to the key database, note the following:

- The value of `dname`, which specifies the Consumer name, can be any attribute from the Consumer subject DN. The Policy Server at the producer site can use its certificate mapping functionality to map the Consumer to a local user directory entry.
- The value for `alias` associated with the private key is the same as the value of the Affiliate Name field. The Attribute Name field is in the Scheme Setup section of SAML Artifact Authentication scheme dialog. The attribute of the Consumer subject DN, represented in the example by the CN value also reflects the Affiliate Name value.

For example, if you entered `CompanyA` as the Affiliate Name, then `alias` is `Company A`. The attribute is `CN=CompanyA, OU=Development, O=CA, L=Waltham, ST=MA, C=US`

- To refer to the existing key store entry, subsequent `keytool` commands use the same `alias`.
- The value for `password` is same as the value of the Password field specified in the Scheme Setup dialog for the SAML Artifact Authentication Scheme.

To create and store a client certificate in the smkeydatabase file at the Consumer

1. Open a command window.
2. If necessary, create a key database by entering:

```
smkeytool -createDB -password fedDB
```

3. Generate a key-pair combination.

For example, to create a private key using the PKCS8 format enter:

```
smkeytool -addPrivKey -alias CompanyA -keyfile idp1pkey.pkcs8 -certfile idp1.crt  
-password smdb
```

This example assumes that you are running smkeytool from the directory where the certificate and key are located, so there are no file paths necessary.

The certificate is now added to the smkeydatabase.

4. Restart the Policy Server to see the smkeydatabase changes immediately.

Protect the Artifact Resolution Service at the Identity Provider

At the Identity Provider Policy Server, configure a policy to [protect the artifact resolution service](#) (see page 238)e. The realm for this policy must use an X.509 client certificate authentication scheme.

How To Protect Resources with a SAML 2.0 Authentication Scheme

Protect target federation resources by configuring a SiteMinder policy that uses the SAML 2.0 authentication scheme.

To protect a federation resource with a SAML authentication scheme:

1. Create a realm that uses the SAML authentication scheme. The realm is the collection of target resources that users request.

Create a realm in one of the following ways:

- Create a unique realm for each authentication scheme already configured.
- [Configure a single target realm](#) (see page 279) that uses a custom authentication scheme to dispatch requests to the corresponding SAML authentication schemes. Configuring one realm with a single target for all Identity Providers simplifies configuration of realms for SAML authentication.

2. After you configure a realm, establish an associated rule and optionally, a response.
3. Group the realm, rule, and response into a policy that protects the target resource.

Important! Each target URL in the realm is also identified in an unsolicited response URL. An unsolicited response is sent from the Identity Provider to the Service Provider, without an initial request from the Service Provider. The unsolicited response contains the target. At the Identity Provider, an administrator must include this response in a link so the Identity Provider can redirect the user to the Service Provider.

Configure a Unique Realm for Each SAML Authentication Scheme

The procedure for configuring a unique realm for each SAML authentication scheme (artifact or profile) follows the standard instructions for creating realms in the FSS Administrative UI.

To create a realm for each SAML authentication scheme

1. Log on to the FSS Administrative UI.
2. Click the System tab.
3. Click Edit, System Configuration, Create Domain.
The Domain dialog opens.
4. Create a policy domain that will contain the realm with the target resources.
5. Create a realm under the policy domain you created in the previous step, noting the following:
 - a. Select the Web Agent protecting the web server where the target federation resources reside for the Agent field.
 - b. Select the SAML authentication scheme for the Authentication Scheme field. This is the SAML scheme that should protect the realm.
6. Create a rule for the realm.

As part of the rule you select a Web Agent action (Get, Post, or Put), which allows you to control processing when users authenticate to gain access to a resource.

Note: Click Help for descriptions of settings and controls, including their respective requirements and limits.

The realm is now configured.

Form the Policy to Protect the Target Resource

After you create the realm, you add it to a policy that protects target federation resources.

Note: The following procedure assumes that a user directory has already been created.

To create a policy for the target federation resources

1. Log on to the FSS Administrative UI.
2. Expand the domain with the target realm.
3. Select the Policies object.

The Policy Properties dialog opens.

4. Configure the policy, using the realm you previously created for federation resources.
5. Save the policy.
6. Exit the FSS Administrative UI.

For more information about creating policies, see the *Policy Server Configuration Guide*.

Configure a Single Target Realm for All Authentication Schemes

To simplify configuration of realms for authentication schemes, create a single target realm for multiple sites generating assertions.

To do this task, set up the following components:

- A single custom authentication scheme
This custom scheme forwards requests to the corresponding SAML or WS-Federation authentication schemes that you already configured for each asserting party.
- A single realm with one target URL

More information:

[Create the Custom Authentication Scheme](#) (see page 280)

[Configure the Single Target Realm](#) (see page 282)

Create SAML Authentication Schemes for the Single Target Realm

Configure the necessary SAML authentication schemes that will be referenced by the custom authentication scheme associated with the single target realm. When you define the custom authentication scheme, you define a parameter that instructs the Policy Server which SAML authentication schemes the custom scheme can apply to resource requests.

To create the SAML authentication scheme

1. Log on to the FSS Administrative UI.
2. Create SAML authentication schemes according to the procedures in this guide for the SAML protocol you are using.
3. Exit the FSS Administrative UI.

More information:

[SAML 1.x Authentication Schemes](#) (see page 256)

[SiteMinder as a Service Provider](#) (see page 347)

Create the Custom Authentication Scheme

A single target realm relies on a specific custom authentication scheme to work properly.

To configure a custom authentication scheme for a single target realm

1. Log on to the FSS Administrative UI.
2. Select the System tab.
3. Select Edit, System Configuration, Create Authentication Scheme.

The Authentication Scheme Properties dialog opens.

4. Complete the fields as follows:

Note: Click Help for descriptions of settings and controls, including their respective requirements and limits.

Name

Enter a descriptive name to indicate this is a custom auth scheme, such as SAML Custom Auth Scheme.

5. Complete the following field in the Scheme Common Setup section:

Authentication Scheme Type

Custom Template

6. Complete the following fields in the Scheme Setup tab

Library

smauthsinglefed

Secret and Confirm Secret

Leave this field blank.

Confirm Secret

Leave this field blank

Parameter

Specify one of the following:

- SCHEMESET=LIST; <saml-scheme1>;<saml_scheme2>

Specifies the list of SAML authentication scheme names to use. If you configured an artifact scheme called artifact_producer1 and POST profile scheme called samlpост_producer2, you will enter these schemes. For example:

SCHEMESET=LIST;artifact_producer1;samlpost_producer2

- SCHEMESET=SAML_ALL;

Specifies all the schemes you have configured. The custom authentication scheme will enumerate all the SAML authentication schemes and find the one with the correct Provider Source ID for the request.

- SCHEMESET=SAML_POST;

Specifies all the SAML POST Profile schemes you have configured. The custom authentication scheme will enumerate the POST Profile schemes and find the one with the correct Provider Source ID for the request.

- SCHEMESET=SAML_ART;

Specifies all the SAML artifact schemes you have configured. The custom authentication scheme will enumerate the artifact schemes and find the one with the correct Provider Source ID for the request.

Enable this scheme for SiteMinder Administrators

Leave unchecked.

7. Click OK to save your changes.

Configure the Single Target Realm

After you configure the authentication schemes, including the custom authentication scheme, you can configure a single target realm for federation resources.

To create the single target realm

1. Log in to the FSS Administrative UI.
2. Select the Domains tab.
3. Select the policy domain you previously created for the single target realm.
4. Select the Realms object and select Edit, Create Realm.

The Realm Properties dialog opens.

5. Enter the following values to create the single target realm:

Note: Click Help for descriptions of settings and controls, including their respective requirements and limits.

Name

Enter a name for this single target realm.

6. Complete the following field in the Resource section:

Agent

Select the SiteMinder Web Agent protecting the web server with the target resources.

Resource Filter

Specify the location of the target resources. Any user requesting a federated resource is be redirected to this location.

For example, /FederatedResources.

7. Select the Protected option in the Default Resource Protection section.
8. Select the previously configured custom authentication scheme in the Authentication Scheme section. This custom authentication uses the `smauthsinglefed` library.

For example, if the custom scheme was named Fed Custom Auth Scheme, you must select this scheme.
9. Click OK.

The single target realm task is complete.

Configure the Rule for the Single Target Realm

Under the single target realm, configure a rule to protect the resources.

1. Select the single target realm.
2. Select Edit, *single target realm*, Create Rule under Realm.

The Rule Properties dialog displays.

3. Enter values for the fields in the dialog.

Note: Click Help for descriptions of settings and controls, including their respective requirements and limits.

4. Click OK.

The rule for the single target realm configuration is created. It can now be used in a policy.

Create a Policy Using the Single Target Realm

Create a policy that references the single target realm. Remember that the single target realm uses the custom authentication scheme that directs requests to the appropriate SAML/WS-Fed authentication scheme.

Note: This procedure assumes that you have already configured the domain, custom authentication scheme, single target realm and associated rule.

To create a policy and add it to an existing domain

1. Log on to the FSS Administrative UI.
2. From the Domains tab, select the Policies object.
3. Click Edit, Create Policy.
The Policy Properties dialog opens.
4. Enter a name and a description of the policy in the General section.
5. Add users to the policy from the Users tab.
6. Add the rule you created for the single target realm from the Rules tab.
The remaining tabs are optional.
7. Click OK.

The policy task is complete.

Chapter 17: Authorize Users with Attributes from an Assertion Query

This section contains the following topics:

[Perform Authorizations with an Attribute Authority](#) (see page 389)

[Flow Diagram for Authorizing a User with User Attributes](#) (see page 392)

[How to Configure an Attribute Authority and a SAML Requester](#) (see page 393)

[Set up the Attribute Authority](#) (see page 393)

[Set up a SAML Requestor to Generate Attribute Queries](#) (see page 396)

Perform Authorizations with an Attribute Authority

The Policy Server authorizes a user with the following types of information:

- Users that are specified in the policy configuration
- Policy expressions
- Active policies
- IP address restrictions
- Time restrictions

The Policy Server also authorizes a user with user attributes that a SAML 2.0 Attribute Authority provides. When a user requests access to a protected resource, the Policy Server, as the authorizing entity, can request more user attributes. The Policy Server evaluates these attributes before granting access to the resource.

The SAML 2.0 Assertion Query/Request profile employs two entities:

- SAML Attribute Authority
- SAML Requester

SAML Attribute Authority

The SAML Attribute Authority relies on an Attribute Service to process a query message and add attributes to an assertion. These assertions contain user attributes that a SAML Requester uses to authorize access to protected resources. The Attribute Service is part of the Federation Web Services application.

When an entity makes a request to an Attribute Authority, the message contains the user attributes that the requester wants to retrieve. The message also contains the Name ID and the Issuer of the request. The Attribute Service uses the NameID to disambiguate the user so it knows what values to return for the requested attributes. The Attribute Service returns a response message that includes an attribute assertion that is wrapped in a SOAP message. This response includes the user attributes.

Note: The user does not need to be authenticated at the Attribute Authority. Also, there is no need for a single sign-on relationship between the Authority and the Requester.

SAML Requester

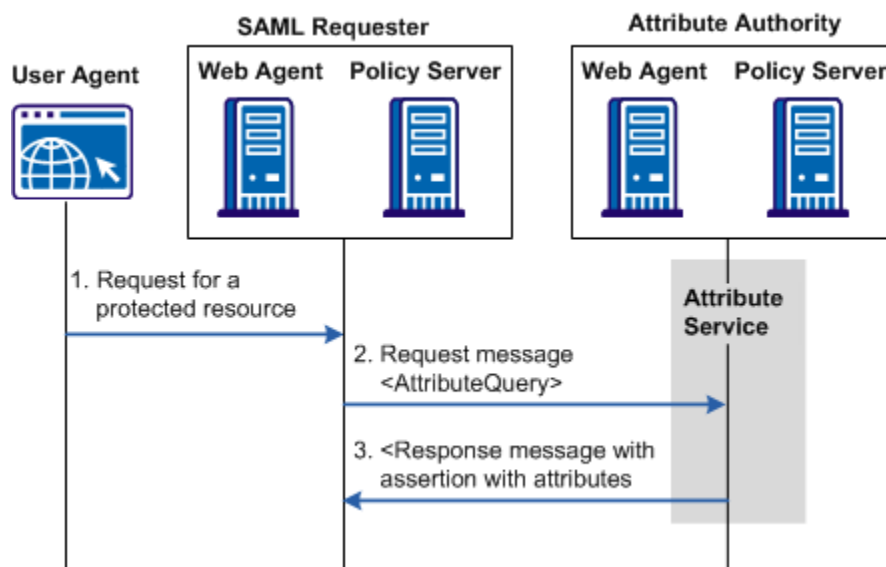
The SAML Requester is a SAML entity that uses the SAML 2.0 Assertion Query/Request profile to request attributes for a user. For SiteMinder, the SAML Requester is not a specific service, but a group of Policy Server features that can produce and process <AttributeQuery> messages. The Requester asks for the user attributes from the Attribute Authority because the protected target resource always resides at the SAML requester. The Requester resolves these attributes into variables that a policy expression uses.

Note: In a SiteMinder federated environment, the SAML Attribute Authority is the Identity Provider and the SAML Requester is the Service Provider. However, this condition does not have to be the case.

To evaluate an authorization request that is based on SAML 2.0 user attributes, add an attribute type named **federation attribute variable** to a policy expression. The policy protecting the target resource uses this variable. Based on the policy variable, the SAML Requester sends a query message to the Attribute Authority. This query message contains the Name ID for the SAML entity for which the attributes are being requested. The SAML Attribute Authority returns a response message containing assertions with the attribute statements.

A user must have a session at the SAML Requester; however, the user does not have to log in or authenticate at the Attribute Authority.

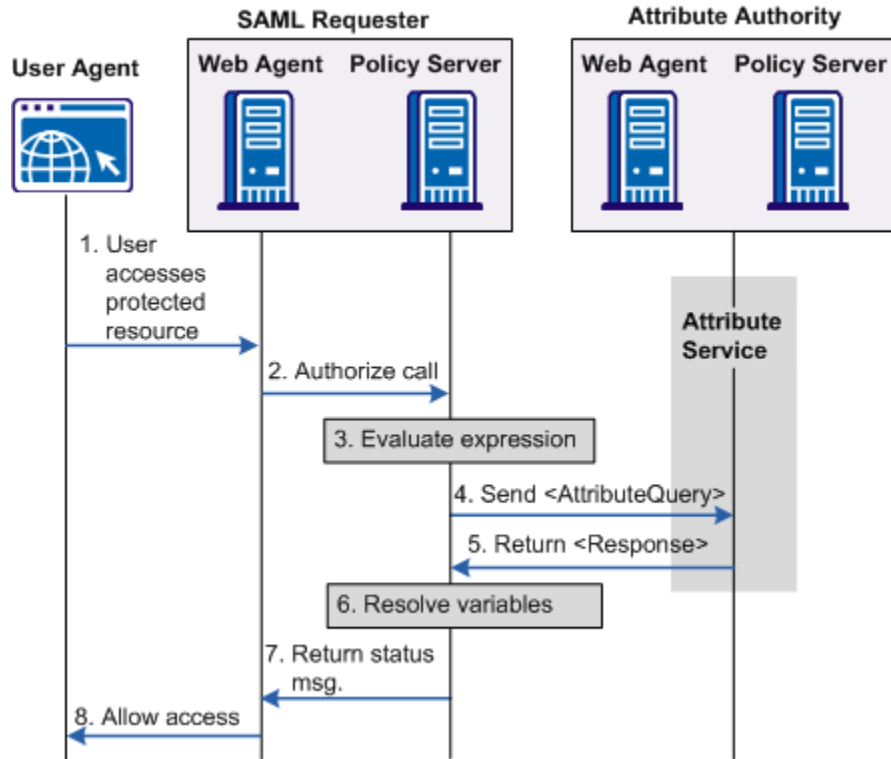
The following figure shows how an attribute query is processed.



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

Flow Diagram for Authorizing a User with User Attributes

The following flow diagram shows the authorization process with an Attribute Authority.



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

The sequence of a user attribute request is as follows:

1. A user accesses a protected resource. The user can log in locally or can be authenticated through a SAML assertion.
2. The Web Agent at the SAML Requester calls the local Policy Server determine whether the user is authorized to access the resource. The policy that protects the resource uses a policy expression for authorization with a federated attribute variable.
3. The Policy Server tries to resolve these variables but cannot. The Policy Server looks up the user in the local user store to obtain the NameID of the user.
4. An attribute query is sent to the AttributeService URL at the Attribute Authority. The AttributeQuery contains the users NameID and the requested attributes.

5. The Attribute Authority returns a SAML response containing an assertion with the requested attributes.
6. The SAML Requester completes the resolution of variables and then evaluates the policy expression.
7. An authorization status message is returned to the Web Agent.
8. Depending on the authorization status, the Web Agent allows or denies access to the requested resource.

How to Configure an Attribute Authority and a SAML Requester

In a SiteMinder context, the Attribute Authority is the Identity Provider.

To configure SiteMinder to act as a SAML Attribute Authority

1. Define a search specification for locating a user. Enter the NameID into the search specification.
2. Configure the back channel across which the Authority sends the response to a query.
3. Define the attributes that are returned in response to a query.
4. Grant users access to the attribute authority service.

In a SiteMinder context, the SAML Requester is the Service Provider.

To configure SiteMinder as a SAML Requester

1. Enable the attribute query functionality.
2. Configure the back channel across which the Requester receives the response from the Authority.
3. Define the list of attributes requested in the attribute query.
4. Configure the federation attribute variables.
5. Configure the NameID for inclusion in the attribute query.

Set up the Attribute Authority

In a SiteMinder context, the Attribute Authority is the Identity Provider with the Attribute Authority service enabled.

Note: You do not need to configure other Identity Provider features, such as single sign-on to have the Identity Provider act as an Attribute Authority.

To configure a SiteMinder Attribute Authority

1. Log on to the FSS Administrative UI.
2. From the appropriate affiliate domain, double-click the Service Provider, acting as the SAML Requester, that requests the user attributes.

The SAML Service Provider Properties dialog opens.

3. Select the Attribute Svc tab.
4. Select Enabled to enable the Attribute Authority feature.
5. (Optional) Modify the value of the Validity Duration. You can accept the default of 60 seconds.

Modify this setting only if you want the assertion to be valid for longer than 60 seconds.

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

6. (Optional) Configure one or both of the signing settings. Neither of these settings are required.

Require Signed Attribute Query

Select this option if you want the Attribute Authority to accept only signed queries from the SAML Requester.

Signing Options

Select one of the options to sign the attribute assertion, the SAML response, both, or neither when they are returned to the SAML Requester.

7. Select a namespace in the User Lookup section and click Edit.

The Attribute Service Namespace Mapping dialog opens.

8. In the Search Specification field, enter a namespace attribute that the authentication scheme uses to search string, then click OK.

Use %s in the entry as the variable that represents the NameID. For example, the NameID has a value of user1. If you specify Username=%s in the Search Specification field, the resulting string is Username=user1. This string is verified against the user store to find the correct record for authentication.

9. Click OK.

You return to the Attribute Svc tab.

10. Click OK to save your changes.
11. Go to [Configure the Attributes at the Attribute Authority](#) (see page 395).

Configure Attributes at the Attribute Authority

When you configure an attribute, you indicate whether the attribute is used as part of a single sign-on request, or to satisfy an attribute query request. The Retrieval Method field in the SAML Service Provider Attribute dialog determines the attributes function.

To use the same attribute for both services, create two attribute statements that use the same Attribute name and variable; however, one attribute uses SSO as the retrieval method and one uses Attribute Services as the retrieval method.

To configure an attribute

1. [Configure Attributes for SSO Assertions](#) (see page 320).

The configuration process for configuring attributes at the Attribute Authority are the same for configuring attributes for single sign-on assertions.

2. Select Attribute Service for the Retrieval Method field in the SAML Service Provider Attribute dialog.

If an attribute query requests this attribute, selecting Attribute Service as the Retrieval Method marks the attribute for inclusion in the attribute assertion.

Configure the Back Channel for the Attribute Authority

If you configure a SAML Attribute Authority, configure a secure backchannel across which the SAML Attribute Authority returns the SAML response to the requester.

Note: This procedure assumes that you have already enabled the Attribute Authority Service.

To configure the backchannel

1. Open the FSS Administrative UI.
2. From the appropriate affiliate domain, double-click the Service Provider that requests the user attributes for authorization.

The SAML Service Provider Properties dialog opens.

3. Select the General tab.
4. Click Configure Backchannel Authentication.

The Backchannel Properties dialog opens.

5. Complete the following fields:

- Password
- Confirm Password

If you configured SAML 2.0 artifact authentication, you have already configured a password for the backchannel. This password can be used for both SSO and the Attribute Authority Service.

6. Click OK to save your entries.

Set up a SAML Requestor to Generate Attribute Queries

For a Service Provider to act as a SAML Requester, configure a SAML 2.0 authentication scheme so that an attribute query can be generated.

To configure the Service Provider as a SAML Requester

1. Log on to the FSS Administrative UI.
2. Display the Authentication Schemes object and double-click an existing SAML 2.0 authentication scheme or create a scheme.

The Authentication Scheme Properties dialog opens.

3. Click Additional Configuration.

The SAML 2.0 Auth Scheme Properties dialog opens.

4. From this dialog, configure fields in the following dialogs:

- Attributes tab
- NameIDs tab
- Backchannel tab

5. Configure a Federation Attribute Variable.

More Information:

[Enable Attribute Queries and Specify Attributes](#) (see page 397)

[Configure the NameID for the Attribute Query](#) (see page 398)

[Configure the Backchannel for the Attribute Query](#) (see page 398)

[Create a Federation Attribute Variable](#) (see page 399)

Enable Attribute Queries and Specify Attributes

To enable the SAML Requester to send an attribute query

1. Log on to the FSS Administrative UI.
2. Access the Authentication Scheme Properties dialog for the SAML 2.0 authentication scheme. The SAML 2.0 authentication scheme protects the resource that is protected based on a user attribute.
3. Click on Additional Configuration.
The SAML 2.0 Auth Scheme Properties dialog opens.
4. Click on the Attributes tab.
5. Click Add.
The Add Attribute dialog opens.
6. Enter values for the following fields:
 - Local Name
 - Attribute Name
 - Name Format

Note: Click Help for a description of fields, controls, and their respective requirements.
7. Click OK to save your changes.
You return to the Attributes dialog.
8. In the Attribute Query section, select Enabled and enter a value for the Attribute Service field.
9. Optionally, select the following check boxes:
 - Sign Attribute Query
 - Require Signed Assertions
 - Get All Attributes
10. Click OK.
The Name IDs tab opens and a message is displayed instructing you to specify an attribute name for the name identifier.
11. [Configure a NameID](#) (see page 398). This NameID configured in the SAML 2.0 Auth.Scheme Properties is included in the attribute query for use by the Attribute Authority.

Configure the NameID for the Attribute Query

When a SAML Requester sends a query message to the Attribute Authority, it includes the NameID of the user whose attributes it is requesting. When you configure the NameID, you are specifying how the SAML Requester obtains the NameID so it can be placed in the attribute query.

To specify a NameID

1. If necessary, select the authentication scheme you want to configure and access the Authentication Scheme Properties dialog.
2. Click Additional Configuration and select the Name IDs tab.
3. Define the following for the NameID:
 - Name ID format
 - Name ID Type
 - Name ID Fields
4. Click OK to save your changes.

If the backchannel is not already configured, you are be prompted to configure it.

Configure the Backchannel for the Attribute Query

The attribute query is sent across a secure backchannel to the Attribute Authority.

Note: Only one backchannel is available between the Service Provider and the Identity Provider. Therefore, the backchannel configuration you define for the attribute query is the same backchannel configuration that is used when you configure the SAML artifact profile.

To configure the backchannel

1. If necessary, access the Authentication Scheme Properties dialog for the SAML requester.
2. Click Additional Configuration.
3. Select the Backchannel tab.
4. Complete the following fields:
 - Authentication
 - SP Name
 - Password
 - Confirm Password

Create a Federation Attribute Variable

To use a federation attribute variable in a policy expression, first create the attribute variable.

To define a federation attribute variable

1. Log on to the FSS Administrative UI.
2. From the list of Domains, expand the policy domain where the variable is added.
3. Expand the Variables list by clicking the plus (+) symbol.
4. Select Federation Attribute Variable then select Edit, Create Variable
The Federation Attribute Variable Properties dialog opens.
5. Complete all the fields in the dialog.
6. Click OK to save the variable.
7. Add this variable to an expression used by a policy that protects a federated resource.

Note: A policy expression can use multiple Federation attribute variables; each variable is tied to a SAML 2.0 authentication scheme. Therefore, a single expression can result in many attribute requests sent to many Attribute Authorities.

Create a Policy Expression with the Federation Attribute Variable

To use Federation attribute variables as part of the authorization process, configure the attribute variable then add it to a policy expression. You then associate this policy expression with the policy protecting the target resource at the SAML requester.

Chapter 18: Configure SiteMinder as an Account Partner

This section contains the following topics:

- [Prerequisites for a SiteMinder Asserting Party](#) (see page 401)
- [How to Configure a SiteMinder Account Partner](#) (see page 402)
- [Add a Resource Partner to an Affiliate Domain](#) (see page 403)
- [Authenticate Users without a SiteMinder Session](#) (see page 404)
- [Select Users for Which Assertions Will Be Generated](#) (see page 406)
- [Specify Name IDs for WS-Federation Assertions](#) (see page 409)
- [Configure Required General Information for WS-Federation](#) (see page 409)
- [Configure Single Sign-on for WS-Federation](#) (see page 411)
- [Customize a WS-Federation Assertion \(optional\)](#) (see page 414)
- [Configure Attributes for WS-Federation Assertions \(optional\)](#) (see page 416)
- [Configure Signout for WS-Federation](#) (see page 420)
- [Set Up Links to Initiate WS-Federation Single Sign-on](#) (see page 422)

Prerequisites for a SiteMinder Asserting Party

For SiteMinder to serve as the asserting party, the following conditions must be met:

- Install the Policy Server.
- Install one of the following:
 - The Web Agent and the Web Agent Option Pack. The Web Agent authenticates users and establishes a SiteMinder session. The Option Pack provides the Federation Web Services application.
 - The SPS federation gateway, which has an embedded Web Agent and has the Federation Web Services application deployed on the embedded Tomcat web server.
- A SAML consumer is set up within the federated network.

The assertions generated by the producer are sent to the consumer. An application at the consumer receives and interprets the assertion. The SAML Affiliate Agent and a system with the SiteMinder Web Agent Option Pack can act as SAML consumers.

How to Configure a SiteMinder Account Partner

SiteMinder, as an Account Partner generates assertions for its business partners, the Resource Partners. To establish a federated partnership, the Account Partner needs information about each Resource Partner. Create a Resource Partner object for each partner. Define how the two entities communicate to pass assertions and to satisfy profiles, such as single sign-on.

To configure SiteMinder to act as an Account Partner

1. Associate the Resource Partner with an affiliate domain.
2. Add a Resource Partner to the affiliate domain.
3. Specify the general identifying information for the Resource Partner.
4. Select users from a user store.

The Account Partner generates assertions for the users you select.

5. Specify the Name ID to include in the assertion.
6. Configure the single sign-on profiles.
7. Complete any [optional configuration tasks](#) (see page 403).

You can save a Resource Partner entity without configuring a complete SSO profile. You cannot actually pass an assertion to the Resource Partner without configuring SSO.

Tips:

- Certain parameter values at the Account Partner and Resource Partner must match for the configuration to work. A list of those parameters is listed in [Configuration Settings that Must Use the Same Values](#) (see page 471).
- To verify that you are using the correct URLs for the Federation Web Services servlets, you can find a list of URLs in [Federation Web Services URLs Used in SiteMinder Configuration](#) (see page 477).

More Information:

[Creating Affiliate Domains](#) (see page 219)

[Add a Resource Partner to an Affiliate Domain](#) (see page 403)

Optional Configuration Tasks for a SiteMinder Account Partner

The following are optional tasks for configuring a Resource Partner:

- Configure single sign-on restrictions:
 - Set IP address restrictions to limit the addresses used to access Resource Partners.
 - Configure time restrictions for Resource Partners.
- Configure attributes for inclusion in assertions.
- Configure sign out.
- Customize a SAML response using the Assertion Generator plug-in.

More Information:

[Specify IP Address Restrictions for Resource Partners \(optional\)](#) (see page 412)

[Set up Time Restrictions for Resource Partner Availability \(optional\)](#) (see page 413)

[Configure Signout for WS-Federation](#) (see page 420)

[Customize a WS-Federation Assertion \(optional\)](#) (see page 414)

Add a Resource Partner to an Affiliate Domain

To identify a Resource Partner as an available consumer of SiteMinder-generated assertions, you add the Resource Partner to an affiliate domain configured at the Account Partner's Policy Server. You then define the Resource Partner's configuration so that the Account Partner can issue security token response messages containing assertions.

To add a Resource Partner to an affiliate domain

1. Log into the FSS Administrative UI.
2. Display the list of domains.
3. Expand the AffiliateDomain and select the Resource Partners object.
4. From the menu bar select Edit, Create Resource Partner.
The Resource Partner Properties dialog opens.
5. Fill in the following fields at the top of the dialog:
 - Name (a unique name)
 - Description
 - Authentication URL

- Use Secure URL
- Application URL

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

6. Check Enabled to enable the Account Partner to recognize the Resource Partner you have identified.

Authenticate Users without a SiteMinder Session

When you add a Resource Partner to an affiliate domain, one of the parameters you are required to set is the Authentication URL parameter.

The Authentication URL points to the `redirect.jsp` file, which is installed at the Account Partner site, where you install the Web Agent Option Pack or SPS federation gateway. A SiteMinder policy must protect the `redirect.jsp` file so that an authentication challenge is presented to users who request a protected Resource Partner resource but do not have a SiteMinder session.

A SiteMinder session is required for the following bindings:

- For users requesting a protected Resource Partner resource
- For single sign-on

A user must have a session, but it does not have to be a persistent session because security token response messages are delivered directly to the Resource Partner site through the browser of the user. The tokens do not have to be stored in the session server.

- For signout

If you enable signout, a persistent session is required. When a user first requests a Resource Partner resource, the session established at that time must be stored in the session server so that the necessary session information is available when signout is later executed.

After a user is authenticated and successfully accesses the `redirect.jsp` file, a session is established. The `redirect.jsp` file redirects the user back to the Account Partner so the request can be processed and the assertion can be delivered to the user.

The procedure for protecting the Authentication URL is the same regardless of the following conditions:

- Web Agent Option Pack is installed on the same system as the Web Agent
- Web Agent Option Pack is installed on an application server with a Web Agent installed on a web server proxy
- Web Agent Option Pack is installed on an application server protected by an Application Server Agent
- SPS federation gateway provides the redirect.jsp file

Create a Policy to Protect the Authentication URL

To create a policy to protect the Authentication URL

1. Log into the FSS Administrative UI.
2. From the System tab, create Web Agents to bind to the realms that you define for the Account Partner Web Server. You can assign unique Agent names for the Web Server at the Account Partner and the Federation Web Services application or use the same Agent name for both.
3. Create a policy domain for the users who want to access Resource Partner resources.
4. From the Users tab, select the users who must have access to the resources that are part of the policy domain.
5. Define a realm for the policy domain with the following values:
 - a. Agent: select the Agent for the Web Server at the Account Partner.
 - b. Resource Filter:

Web Agents v5.x QMR 4 and later, and SPS federation gateway enter:
`/siteminderagent/redirectjsp/`

Web Agents v5.x QMR 1, 2, or 3, enter:
`/affwebservices/redirectjsp/`

The resource filter, `/siteminderagent/redirectjsp/` is an alias, set up automatically by the Federation Web Services application. It is a reference to the following:
 - For a Web Agent:
`web_agent_home/affwebservices/redirectjsp`
 - For an SPS federation gateway:
`sps_home/secure-proxy/Tomcat/webapps/affwebservices/redirectjsp`
- c. For the remaining settings, accept the defaults or modify as needed.

6. Click OK to save the realm.
7. Create a rule for the realm. In the Resource field, accept the default value, the asterisk (*), to protect all resources for the realm. Select the Web Agent actions GET, POST, and PUT as the allowed actions.
8. Create a policy for the Web Server at the Account Partner that includes the rule created in the previous step.

Select Users for Which Assertions Will Be Generated

When you configure a Resource Partner, you include a list of users and groups for which the WS-Federation Assertion Generator generates SAML assertions.

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

To specify users and groups that have access to Resource Partner resources

1. Log in to the FSS Administrative UI.
2. Access the Resource Partner Properties dialog and select the Users tab.
If the associated affiliate domain contains more than one user directory, the directories appear as subordinate tabs on the Users tab.
3. Click the Add/Remove button.
The Users/Groups dialog opens.
4. To add users, select an entry from the Available Members list and click the Left Arrow button, which points to the Current Members list.
Reversing the procedure removes users from the Current Members list.
 - You can select multiple entries by holding the CTRL or SHIFT key and clicking entries in one of the Members lists. When you select multiple entries and click one of the Arrow buttons, the FSS Administrative UI moves all of the selected entries.
 - Individual users are not displayed automatically. However, you can use the Search utility to find a specific user within one of the listed groups. Different types of user directories must be searched differently.
5. Click OK to save your changes.

Excluding a User or Group from Resource Partner Access

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

To exclude a user or group from access to the resources of a Resource Partner

1. In the Users/Groups dialog, select a user or group from the Current Members list.
2. To exclude the selected user or group, click Exclude.

The symbol to the left of the item in the Current Members list indicates whether the user or group is excluded from the Resource Partner.

3. Click OK.

Allow Nested LDAP Groups Resource Partner Access

LDAP user directories can contain groups nested in other groups. In complex directories, large amounts of user information can be organized in a nested hierarchy.

When you enable a Resource Partner to search for users in nested groups, the Policy Server searches any subset group you add to a policy. If you do not enable nested groups, the Policy Server only searches the single group you specify for the Resource Partner.

To allow the Resource Partner to search nested groups

1. From the Users tab, select the Allow Nested Groups check box to enable nested groups searching for the Resource Partner.
2. If the associated affiliate domain contains more than one user directory, the directories appear as tabs on the User tab.

Add Users by Manual Entry for Resource Partner Access

From the Users/Groups dialog, you can use the Manual Entry option to add users who can access the Resource Partner resources.

To add a user by manual entry

1. In the Manual Entry section, do one of the following:
 - For LDAP directories, enter a valid DN
For each DN, you can select an action from the Action drop-down list, as follows:
Search Users--the LDAP search is limited to matches in user entries.
Search Groups--the LDAP search is limited to matches in group entries.
Search Organizations--the LDAP search is limited to matches in organization entries.
Search Any Entry--the LDAP search is limited to matches in user, group, and organization entries.
Validate DN--the LDAP search locates this DN in the directory.
 - For Microsoft SQL Server, Oracle and WinNT directories, enter a user name in the Manual Entry field.
For a Microsoft SQL Server or Oracle, you can enter a SQL query instead. For example:

```
SELECT NAME FROM EMPLOYEE WHERE JOB ='MGR';
```


The Policy Server performs the query as the database user-specified in the Username field of the Credentials and Connection tab for the user directory. When constructing the SQL statement for the Manual Entry field, familiarize yourself with the database schema for the user directory. For example, if you are using the SmSampleUsers schema and want to add specific users, you could select a user entry from the SmUser table.
- Note:** For an LDAP directory, you can enter all in the Manual Entry field to add all directory entries to the Resource Partner.
2. Click Add to Current Members.
 3. Click OK to save your changes.

Specify Name IDs for WS-Federation Assertions

A name ID names a user in an assertion in a unique way. The value you configure in the FSS Administrative UI will be included in the assertion sent to the Resource Partner.

The format of the name ID establishes the type of content used for the ID. For example, the format might be the User DN so the content would be a uid.

Configure a Name ID for a WS-Federation Assertion

To configure a name ID

1. Log in to the FSS Administrative UI and access the Resource Partner entry you want to configure.
2. Select the Name IDs tab on the Resource Partner Properties dialog box.
3. Select the Name ID Format.

For a description of each format, see Section 8.3 of the *Assertions and Protocols for the OASIS Security Assertion Markup Language (SAML) V2.0* specification (sstc-saml-core-2.0-cd-04.pdf).

4. Choose the Name ID Type from the following options:

- Static value
- User attribute
- DN attribute (with or without nested groups)

The contents of the Name ID Fields group box change according to the Name ID Type selected.

5. Complete the fields for the selected Name ID Type.

Configure Required General Information for WS-Federation

Select the General tab to configure required items, such as the ID of the Resource Partner and Account Partner.

To configure the general settings

1. Log in to the FSS Administrative UI.
2. Open the Resource Partner Properties dialog.
3. Select the General tab.

4. Fill-in values for the following required fields:

Resource Partner ID

Specifies a URI that uniquely identifies the Resource Partner, such as, rp.example.com.

Account Partner ID

Specifies a URI that uniquely identifies the Account Partner, such as ap-ca.com. This becomes the Issuer field in the SAML assertion.

Skew Time

Specifies the number of seconds (as a positive integer) to be subtracted from the current time to account for Resource Partners that have clocks that are not synchronized with the Policy Server acting as an Account Partner.

For single sign-on, the value of the Skew Time and the single sign-on validity duration (Validity Duration field on the SSO tab) determine how long an assertion is valid. Review how the [assertion validity](#) (see page 298) is calculated to understand more about the skew time.

5. For debugging purposes only, you can temporarily disable all signature processing (both signing and verification of signatures) by selecting the Disable Signature Processing checkbox.

Important! By default, signature processing is enabled because it is required by the WS-Federation Passive Requester profile for single sign-on; therefore, it *must* be enabled in a production environment.

More Information:

[Set the Skew Time WS-Federation Single Sign-on](#) (see page 410)

Set the Skew Time WS-Federation Single Sign-on

In the Skew Time field on the General tab, enter the difference, in seconds, between the system clock at the Account Partner and the system clock at the Resource Partner.

For single sign-on, the values of the Validity Duration (set on the SSO tab) and Skew Time (set on the General tab) instruct how the WS-Federation Assertion Generator calculates the total time that an assertion is valid. In the assertion document, the beginning and end of the validity interval is represented by the NotBefore and NotOnOrAfter values.

To determine the beginning of the validity interval, the assertion generator takes the system time when the assertion is generated and sets the IssueInstant value in the assertion according to this time. It then subtracts the Skew Time value from the IssueInstant value. The resulting time becomes the NotBefore value.

To determine the end of the validity interval, the assertion generator adds the Validity Duration value and the Skew Time together. The resulting time becomes the NotOnOrAfter value. Times are relative to GMT.

For example, an assertion is generated at the Account Partner 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 prior to the time the assertion was generated and ends 90 seconds afterward.

Configure Single Sign-on for WS-Federation

The Resource Partner and the Account Partner exchange user information, session information and Account Partner information, in an assertion document sent in a security token response message. When you configure single sign-on at the Account Partner, you determine how the Account Partner delivers an assertion to a Resource Partner.

To set-up single sign-on at the Account Partner

1. Log in to the FSS Administrative UI.
2. Select the Resource Partner you want to configure.
3. Open the Resource Partner Properties dialog.
4. Select the SSO tab.
5. Fill in entries for the following fields on this tab:
 - Authentication Method
 - Validity duration
 - Security Token Consumer Service
 - Authentication Level
6. Optionally, configure policy restrictions based on IP address or time by clicking on Restrictions and completing the appropriate fields.

More Information:

[Customize a WS-Federation Assertion \(optional\)](#) (see page 414)

Set the Authentication Scheme Protection Level

The WS-Federation Assertion Generator creates an assertion based on a user session. The user associated with the session has been authenticated at a particular authentication scheme protection level. This means that you can control which users an assertion is generated for based on the protection level at which they authenticated.

Users are authenticated at different protection levels. Therefore, the assertions generated should be for users who authenticated at the required level. Failure to adhere to the protection level may compromise the federated environment's security because the assertions may misrepresent the authentication level at which a user actually authenticated.

Specify IP Address Restrictions for Resource Partners (optional)

The FSS Administrative UI allows you to specify an IP address, range of IP addresses, or a subnet mask of the Web server on which a user's browser must be running for the user to access a Resource Partner. If IP addresses have been specified for a Resource Partner, only users who access the Resource Partner from the appropriate IP addresses are accepted.

To specify IP addresses

1. Log in to the FSS Administrative UI.
2. Select the Resource Partner you want to configure.
3. Open the Resource Partner Properties dialog.
4. Select the SSO tab, then click on Restrictions.
5. Click Add.

The Add an IP Address dialog box opens.

6. Select one of the following radio buttons to indicate the type of IP address value you are adding:

Note: If you do not know the IP address, but you have a domain name for the address, click on the DNS Lookup button. In the DNS Lookup dialog box, enter a fully qualified host name in the Host Name field and click OK.

- Single Host--specifies a single IP address that hosts the user's browser. If you specify a single IP address, the Resource Partner can only be accessed by users from the specified IP address.
- Host Name--specifies a Web server using its host name. If you specify a host name, the Resource Partner is only accessible to users who access it from the specified host.

- Subnet Mask--specifies a subnet mask for a Web server. If you specify a subnet mask, the Resource Partner is only accessible to users who access the Resource Partner resources from the specified subnet mask.

When you select this radio button, the Add an Address and Subnet Mask dialog opens. Use the Left and Right arrow buttons, or click and drag the slider bar to select a subnet mask.

- Range--specifies IP address range. If you specify a range of IP addresses, the Resource Partner only permits users who access the Resource Partner resources from one of the IP addresses in the range of addresses. You enter a starting (FROM) and ending (TO) addresses to determine the range.

7. Click OK to save your configuration.

Set up Time Restrictions for Resource Partner Availability (optional)

You can specify time restrictions that indicate a Resource Partners's availability. When you add a time restriction, the Resource Partner functions only during the period specified. If a user attempts to access a resource outside of that period, the Account Partner does not produce assertions.

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

To specify a time restriction

1. Log in to the FSS Administrative UI.
2. Select the Resource Partner you want to configure.
3. Open the Resource Partner Properties dialog.
4. Select the SSO tab, then click Restrictions.
5. In the Time Restrictions group box, click Set.

The Time dialog box opens. This dialog box is identical to the Time Restrictions dialog box used for rule objects.

6. Click OK.

Customize a WS-Federation Assertion (optional)

The WS-Federation Assertion Generator produces SAML assertions. Assertions are the basis for user authentication in a federated environment. You can customize the content of the SAML assertion by configuring an Assertion Generator plug-in. Using this plug-in, you can modify the assertion content for your business agreements between partners and vendors.

To use the WS-Federation Assertion Generator plug-in

1. Implement the plug-in class.
A sample class, `AssertionSample.java`, can be found in `sdk/samples/assertiongeneratorplugin`.
2. Configure the Assertion Generator plug-in from the Advanced tab of the Resource Partner Properties dialog.

Note: Specify an Assertion Generator plug-in for each Resource Partner.

- a. In the Full Java Class Name field, enter the Java class name of the plug-in.

For example, `com.mycompany.assertiongenerator.AssertionSample`

A sample plug-in is included in the SDK. You can view the sample assertion plug-in at `sdk/samples/assertiongeneratorplugin`.

- b. Optionally, in the Parameters field, enter the string that gets passed to the plug-in as a parameter at run time.

The string can contain any value; there is no specific syntax to follow.

Note: For reference information about the WS-Federation Assertion Generator plug-in, see the `AssertionGeneratorPlugin` interface in the *Javadoc Reference*. This information applies to the WS-Federation Assertion Generator and the SAML Assertion Generator. For overview and conceptual information, see the *SiteMinder Programming Guide for Java*.

Implement the AssertionGeneratorPlugin Interface

The first step in creating a custom assertion generator plug-in is to implement the `AssertionGeneratorPlugin` interface.

Follow these steps:

1. Provide a public default constructor method that contains no parameters.
2. Provide code so that the implementation is stateless. Many threads must be able to use a single plug-in class.

3. Implement methods in the interface to satisfy your requirements.

The implementation must include a call to the `customizeAssertion` methods. You can overwrite the existing implementations. See the following sample classes for examples:

SAML 1.x/WS-Federation

`AssertionSample.java`

SAML 2.0

`SAML2AssertionSample.java`

The sample classes are located in the directory `/sdk/samples/assertiongeneratorplugin`.

Note: The contents of the parameter string that your implementation passes into the `customizeAssertion` method is the responsibility of the custom object.

Deploy the Assertion Generator Plug-in

After you have coded your implementation class for the `AssertionGeneratorPlugin` interface, compile it and verify that SiteMinder can find your executable file.

To deploy the assertion generator plug-in

1. Compile the assertion plug-in Java file.

Compilation requires the following .jar files, which are installed with the Policy Server:

- `policy_server_home/bin/jars/SmJavaApi.jar`
- `policy_server_home/bin/thirdparty/xercesImpl.jar`
- `policy_server_home/bin/endorsed/xalan.jar`

2. In the `JVMOptions.txt` file, modify the `-Djava.class.path` value so it includes the classpath for the plug-in. This modification enables the plug-in to be loaded with the modified classpath. Locate the `JVMOptions.txt` file in the directory `installation_home\siteminder\config`.

Note: Do not modify the classpath for `xercesImpl.jar`, `xalan.jar`, or `SMJavaApi.jar`.

3. Enable the plug-in.

Enable the Assertion Generator Plug-in (WS-Federation)

After writing an assertion generator plug-in and compiling it, enable the plug-in by configuring settings in the FSS Administrative UI. The UI parameters let SiteMinder 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 in to the FSS Administrative UI.
2. Navigate to the Resource Partner Properties dialog and access the Advanced tab.
3. Complete the following fields:

Full Java Class Name

Specify a Java class name for an existing plug-in.

Parameter

Specify a string of parameters that is passed to the plug-in specified in the Full Java Class Name field.

Note: Instead of specifying the assertion plug-in class and its parameters through the Administrative UI, you can use the Policy Management API (C or Perl) to integrate the plug-in. For more information, see the *SiteMinder Programming Guide for C* or the *SiteMinder Programming Guide for Java*.

4. Restart the Policy Server.

Restarting the Policy Server verifies that the latest version of the assertion plug-in is picked up after being recompiled.

Configure Attributes for WS-Federation Assertions (optional)

Attributes can provide information about a user requesting access to a Resource Partner resource. An attribute statement passes user attributes, DN attributes, or static data from the Account Partner to the Resource Partner in a SAML assertion. Any configured attributes are included in the assertion in one <AttributeStatement> element or the <EncryptedAttribute> element in the assertion.

Note: Attribute statements are not required in an assertion.

Servlets, web applications, or other custom applications use attributes to display customized content or enable other custom features. When used with web applications, attributes can implement fine-grained access control by limiting user activity at the Resource Partner. For example, you can send an attribute variable named Authorized Amount set to a maximum dollar amount. The amount is the limit that the user can spend at the Resource Partner.

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

Attributes can be made available as HTTP Headers or HTTP Cookies.

The HTTP headers and HTTP cookies have size restrictions that assertion attributes cannot exceed. The size restrictions are as follows:

- For HTTP headers, SiteMinder can send an attribute in a header up to the web server size limit for a header. Only one assertion attribute per header is allowed. See the documentation for your web server to determine the header size limit.
- For HTTP cookies, SiteMinder can send a cookie up to the size limit for a cookie. Each assertion attribute is sent as its own cookie. The cookie size limit is browser-specific, and that limit is for all attributes being passed to the application, not for each attribute. See the documentation for your web browser to determine the cookie size limit.

Configure Assertion Attributes for WS-Federation

To configure assertion attributes

1. Log on to the FSS Administrative UI.
2. In the Resource Partner Properties dialog, click on the Attributes tab.
3. Click Create.
The Resource Partner Attribute dialog box opens.
4. From the Attribute drop down list, select the name format identifier, which is specified by the <NameFormat> attribute in the <Attribute> element of an assertion attribute statement. This value classifies the attribute name so that the Resource Partner can interpret the name.

The options are:

- EmailAddress
- UPN
- CommonName
- Group
- NameValue

For more information on these options, refer to the WS-Federation specification.

5. On the Attribute Setup tab, select one of the following radio buttons:

Note: The radio button selection determines the available fields in the Attribute Fields group box.

Static

Returns data that remains constant.

Use a static attribute to return a string as part of a SiteMinder response. This type of response can be used to provide information to a Web application. For example, if a group of users has specific customized content on a Web site, the static response attribute, `show_button = yes`, could be passed to the application.

User Attribute

Returns profile information from a user's entry in a user directory.

This type of response attribute returns information associated with a user in a directory. A user attribute can be retrieved from an LDAP, WinNT, or ODBC user directory.

For the Policy Server to return values from user directory attributes as response attributes, the user directories must be configured in the User Directory dialog box.

DN Attribute

Returns profile information from a directory object in an LDAP or ODBC user directory.

This type of attribute is used to return information associated with directory objects to which the user is related. Groups to which a user belongs, and Organizational Units (OUs) that are part of a user DN, are examples of directory objects whose attributes can be treated as DN attributes.

For example, you can use a DN attribute to return a company division for a user, based on the user's membership in a division.

Note: For the Account Partner to return an attribute containing DN attributes values, the user directories must be configured in the User Directory dialog box.

If you select the DN Attribute radio button, you may also select the Allow Nested Groups check box. Selecting this check box allows SiteMinder to return an attribute from a group that is nested in another group specified by a policy. Nested groups often occur in complex LDAP deployments.

Note: For attributes from an LDAP user store, you can add multi-valued user attributes to an assertion.

6. Optionally, if the attribute is retrieved from an LDAP user directory that contains nested groups (groups that contain other groups), and you want the Policy Server to retrieve DN attributes from the nested groups, select the Allow Nested Groups check box in the Attribute Kind group box.
7. Complete the necessary fields for you Attribute Kind and save the changes.

Specify the Maximum Length of Assertion Attributes

The maximum length for user assertion attributes is configurable. To modify the maximum length of assertion attributes, change the settings in the EntitlementGenerator.properties file.

Note: The property name in the file is specific to the protocol you are configuring.

Follow these steps:

1. On the system where the Policy Server is installed, navigate to `policy_server_home\config\properties\EntitlementGenerator.properties`.
2. Open the file in a text editor.
3. Adjust the maximum user attribute length for the protocols in use in your environment. The settings for each protocol are as follows:

WS-Federation

Property Name:

`com.netegrity.assertiongenerator.wsfed.MaxUserAttributeLength`

Property Type: Positive Integer value

Default Value: 1024

Description: Indicates the maximum attribute length for WS-FED assertion attributes.

SAML 1.x

Property Name:

`com.netegrity.assertiongenerator.saml1.MaxUserAttributeLength`

Property Type: Positive Integer value

Default Value: 1024

Description: Indicates the maximum attribute length for SAML1.1 assertion attributes.

SAML 2.0

Property Name:

com.netegrity.assertiongenerator.saml2.MaxUserAttributeLength

Property Type: Positive Integer value

Default Value: 1024

Description: Indicates the maximum attribute length for SAML2.0 assertion attributes

4. Restart the Policy Server after any change to these parameters.

Use a Script to Create a New Attribute

The Advanced tab of the Resource Partner Attribute dialog contains the Script field. This field displays the script that SiteMinder generates based on your entries in the Attribute Setup tab. You can copy the contents of this field and paste them into the Script field for another response attribute.

Note: If you copy and paste the contents of the Script field for another attribute, select the appropriate option in the Attribute Kind section of the Attribute Setup tab.

Configure Signout for WS-Federation

Signout is the process of a user being logged out of all sessions for the browser that initiated the logout. Signout does not necessarily end all sessions for a user. For example, if the user has two browsers open, that user can establish two independent sessions. Only the session for the browser that initiates the signout is terminated at all federated sites for that session. The session in the other browser is still active.

A user can initiate a signout request from an Account Partner or a Resource Partner. The request is triggered by clicking a link pointing to the appropriate servlet.

Note: SiteMinder only supports the WS-Federation Passive Request profile for signout.

Enable Signout

By configuring the settings in the Signout section, you are informing the Account Partner how the Resource Partner supports signout.

If you enable signout, you must also:

- Enable the session store at the Account Partner using the Policy Server Management Console.

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

- Configure persistent sessions for the realm with the protected resources at the Resource Partner.

For information about realms, see the *Policy Server Configuration Guide*.

To configure signout

1. Navigate to the SAML Profiles page for the Resource Partner you want to configure.
2. In the Signout section, select Enable Signout.
3. Enter values for the following URL fields:
 - Signout Cleanup URL
 - Signout Confirm URL

These fields must each have an entry that starts with https:// or http://.

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

4. Click OK.

Validate Signout Requests that are Digitally Signed

The WS-Federation Passive Requester profile requires signature processing. Enable signature processing in a production environment. SiteMinder acting as a Resource Partner always signs WS-Federation signout requests. No configuration in the Administrative UI is required. The only required step is to add the private key/certificate pair to the certificate data store for the SiteMinder Resource Partner.

Important! For debugging purposes only, you can temporarily disable all signature processing on the General dialog.

For the Account Partner to validate signout request signatures, some configuration is required.

To enable validation

1. Add the public key to the certificate data store at the Account Partner.

The public key must correspond to the private key/certificate pair that the Resource Partner used to do the signing.

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

2. Navigate to the SAML Profiles page for the Resource Partner object you are configuring.
3. Select Enable Signout in the Signout section.

By selecting this check box, signout is enabled and the Account Partner validates the signature of the signout request.

Set Up Links to Initiate WS-Federation Single Sign-on

You can set up links to initiate single sign-on from either side of a WS-Federation network.

Initiate Single Sign-on at the Account Partner

A user can visit the Account Partner before going to the Resource Partner. If the user goes to the Account Partner first, a link must generate an HTTP Get request. The hard-coded link points to the Single Sign-on Service of the Account Partner. The request contains the RP Provider ID and optionally other parameters.

The syntax for the link to the Single Sign-on Service is as follows:

```
https://ap_server:port/affwebservices/public/wsfedsso?wa=wsigin1.0&wtrealm=RP_ID
```

ap_server:port

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

RP_ID

Resource Partner identity

Initiate Single Sign-on at the Resource Partner

When a user starts at the Resource Partner to initiate single sign-on, typically the user selects from a list of Account Partners. The site selection page is in an unprotected realm.

The link on the site selection page points to the Single Sign-on Service at an Account Partner. After the link is selected, the Resource Partner redirects the user to the Account Partner to get the assertion.

Chapter 19: Configure SiteMinder as a Resource Partner

This section contains the following topics:

- [SAML 1.x Authentication Scheme Prerequisites](#) (see page 425)
- [How to Configure a SiteMinder Resource Partner](#) (see page 426)
- [WS-Federation Authentication Scheme Overview](#) (see page 427)
- [Configure the WS-Federation Authentication Scheme](#) (see page 428)
- [Locate User Records for Authentication](#) (see page 429)
- [Configure WS-Federation Single Sign-on at the Resource Partner](#) (see page 431)
- [Create a Custom WS-Federation Authentication Scheme \(optional\)](#) (see page 432)
- [Implement WS-Federation Signout](#) (see page 433)
- [Customize Assertion Processing with the Message Consumer Plug-in](#) (see page 434)
- [Redirect Users After Failed Authentication Attempts](#) (see page 437)
- [Supply SAML Attributes as HTTP Headers](#) (see page 438)
- [How To Protect a Target Resource with a WS-Federation Authentication Scheme](#) (see page 443)

SAML 1.x Authentication Scheme Prerequisites

There are several prerequisites you must fulfill before configuring a SiteMinder relying partner.

- Install the Policy Server.
For installation instructions, refer to the Policy Server Installation Guide.
- Install one of the following
 - The Web Agent and the Web Agent Option Pack. The Web Agent authenticates users and establishes a SiteMinder session. The Option Pack provides the Federation Web Services application. Be sure to deploy the FWS application on the appropriate system in your network.
 - The SPS federation gateway, which has an embedded Web Agent and has the Federation Web Services application on the embedded Tomcat web server.

For more information, see the *Web Agent Option Pack Guide* or the *Secure Proxy Server Administration Guide*.

- Set up a key database for each Policy Server that is responsible for signing, verification or both. Import private keys and certificates for functions that require verification and encrypting of messages.

The key database is a flat-file key and certificate database that lets you manage and retrieve keys and certificates required to sign and validate SAML responses used with SAML POST profile authentication.

- An asserting partner is set up within the federated network.

How to Configure a SiteMinder Resource Partner

Configuring SiteMinder as a WS-Federation Resource Partner involves the following tasks:

1. Complete the WS-Federation authentication scheme prerequisites.
2. Select the authentication scheme type and assign it a name.
3. Specify the namespace for users being authenticated with the WS-Federation authentication scheme.
4. Configure single sign-on.
5. Complete any option configuration tasks.

Configure an authentication scheme for each Account Partner that is a federation partner and generates assertions. Bind each scheme to a realm, which includes the URLs of the target resources that users request. You can do this task on a per Account Partner basis or can create a single custom authentication scheme and single realm. Protect these resources with a SiteMinder policy.

Tips:

Certain parameter values at the Account Partner and Resource Partner are required to match for the configuration to work. A list of those parameters is located in [Configuration Settings that Must Use the Same Values](#) (see page 471).

Verify that you are using the correct URLs for FWS servlets. See the list in [Federation Web Services URLs Used in SiteMinder Configuration](#) (see page 477).

More Information:

[How To Protect a Target Resource with a WS-Federation Authentication Scheme](#) (see page 443)

Optional Tasks to Configure a SiteMinder Resource Partner

The optional tasks for configuring SiteMinder as a Resource Partner include:

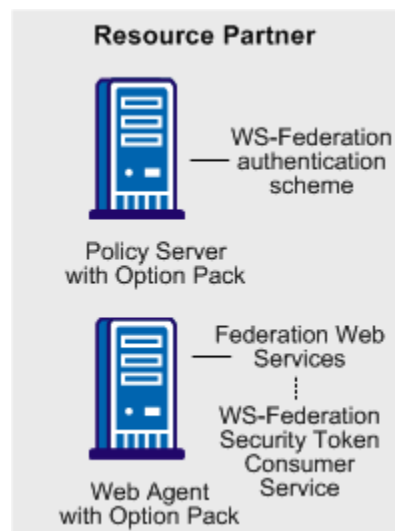
- Customize assertions using the Message Consumer Plug-in.
- Redirect failed authentication attempts.

WS-Federation Authentication Scheme Overview

If you purchased the Policy Server or SPS federation gateway, any SiteMinder site can consume a WS-Federation <RequestSecurityTokenResponse> message and use the assertion in the response to authenticate and authorize users. If you have sites in your federated network that have user stores, you can use WS-Federation authentication.

The WS-Federation authentication scheme lets a Resource Partner authenticate a user. It enables cross-domain single sign-on by consuming a SAML assertion and establishing a SiteMinder session. After the user is identified, the Resource Partner site can authorize the user for specific resources.

A site can be both a WS-Federation Resource Partner and Account Partner.



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

The WS-Federation authentication scheme is configured at the Resource Partner-side Policy Server and is invoked by the WS-Federation Security Token Consumer Service. The Security Token Consumer Service is a component of the Federation Web Services application and is installed on the Resource Partner-side Web Agent. This service obtains information from the WS-Federation authentication scheme at the Policy Server and uses that information to extract the necessary information from the assertion to authenticate a user.

The SAML assertion becomes the user credentials to login to the Policy Server at the Resource Partner site. The user is authenticated and authorized, and if authorization is successful, the user is redirected to the target resource.

Configure the WS-Federation Authentication Scheme

The configuration of the WS-Federation authentication scheme provides information about the Account Partner that generates the assertion for the Resource Partner and instructs how the Resource Partner supports the authentication process.

To configure the common setup and scheme setup

1. Complete the authentication scheme prerequisites.
2. Log in to the FSS Administrative UI.
3. From the menu bar, select Edit, System Configuration, Create Authentication Scheme.

The Authentication Scheme Properties dialog opens.

4. From the Authentication Scheme Type drop-down list, select WS-Federation Template.

The contents of the SiteMinder Authentication Scheme dialog change for the scheme.

5. Configure the scheme common setup section by entering values for the fields.

Note: Click Help for descriptions of settings and controls, including their respective requirements and limits.

6. Configure the scheme setup by entering values for the following fields:
 - Resource Partner ID
 - Account Partner ID
 - Skew Time
 - Alias (required if signature processing enabled)
7. Verify that the Disable Signature Processing option is set appropriately for single sign-on.

Important! For debugging purposes only, you can temporarily disable all signature processing (both signing and verification of signatures) by enabling the Disable Signature Processing option.

After you configure an authentication scheme, associate the scheme with a realm that contains the resource you want to protect.

More Information:

[How To Protect a Target Resource with a WS-Federation Authentication Scheme](#) (see page 443)

Locate User Records for Authentication

When you configure an authentication scheme, you define a way for the authentication scheme to look up a user in the local user store. After the correct user is located, the system generates a session for that user. Locating the user in the user store is the process of disambiguation. How SiteMinder disambiguates a user depends on the configuration of the authentication scheme.

For successful disambiguation, the authentication scheme first determines a LoginID from the assertion. By default, the LoginID is extracted from the Name ID value in the assertion. You can also obtain the LoginID by specifying an Xpath query.

After the authentication scheme determines the LoginID, SiteMinder checks if a search specification is configured for the authentication scheme. If no search specification is defined for the authentication scheme, the LoginID is passed to the Policy Server. The Policy Server uses the LoginID together with the user store search specification to locate the user. For example, imagine that the LoginID value is Username and the LDAP search specification is set to the uid attribute. The Policy Server uses the uid value (Username=uid) to search for the user.

If a search specification is configured for the authentication scheme, the LoginID is not passed to the Policy Server. Instead, the search specification is used to locate a user.

The disambiguation process involves two steps:

1. Obtain the LoginID by the default behavior or by using an Xpath query.
2. Locate the user in the user store by the default behavior or with a search specification.

Note: The use of Xpath and the search specification are optional.

Obtain a LoginID for a WS-Federation User

You can find the LoginID in two ways:

- Relying on the default behavior, where the LoginID is extracted from the NameID in the assertion. This option requires no configuration.
- Using an Xpath query to find the LoginID in place of the default behavior.

To use an Xpath query to locate a user record

1. From the Authentication Scheme Properties dialog, click Additional Configuration.

The WS-Federation Auth Scheme Properties dialog opens.

2. Select the Users tab.

The Users tab specifies who has access to protected resources at the Resource Partner. Access to resources at the Resource Partner is based on SiteMinder policies.

3. Enter an Xpath query that the authentication scheme uses to obtain a LoginID.

Note: Click Help for descriptions of settings and controls, including their respective requirements and limits.

Xpath queries must not contain namespace prefixes. The following example is an invalid Xpath query:

```
/saml:Response/saml:Assertion/saml:AuthenticationStatement/  
saml:Subject/saml:NameIdentifier/text()
```

The valid Xpath query is:

```
//Response/Assertion/AuthenticationStatement/Subject/  
NameIdentifier/text()
```

4. Click OK to save your configuration changes.

Use a Search Specification to Locate a WS-Federation User

You can use a search specification to locate a user record in place of the default behavior of the LoginID being passed to the Policy Server to locate the user.

To locate a user with a search specification

1. From the Authentication Scheme Properties dialog, click Additional Configuration.
The WS-Federation Auth Scheme Properties dialog opens.
2. Select the Users tab.
3. Select a namespace to match the search specification to and click Edit.
The SiteMinder Authentication Scheme Namespace Mapping dialog opens.
4. In the Search Specification field, enter the attribute that the authentication scheme uses to search a namespace, then click OK. Use %s in the entry as a variable representing the LoginID.

Note: Click Help for descriptions of settings and controls, including their respective requirements and limits.

For example, the LoginID has a value of user1. If you specify Username=%s in the Search Specification field, the resulting string is Username=user1. This string is verified against the user store to find the correct record for authentication.

5. Click OK to save your configuration changes.

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

The SSO tab configures the WS-Federation single sign-on binding for authentication. This tab also enforces single use assertion policy to prevent the replaying of a valid assertion.

Part of the single sign-on configuration is defining the Redirect Mode setting. The Redirect Mode specifies how Federation Security Services sends assertion attributes, if available, to the target application. You can send assertion attributes as HTTP Headers or HTTP cookies.

The HTTP headers and HTTP cookies have size restrictions that assertion attributes cannot exceed. The size restrictions are as follows:

- For HTTP headers, SiteMinder can send an attribute in a header up to the web server size limit for a header. Only one assertion attribute per header is allowed. See the documentation for your web server to determine the header size limit.
- For HTTP cookies, SiteMinder can send a cookie up to the size limit for a cookie. Each assertion attribute is sent as its own cookie. The cookie size limit is browser-specific, and that limit is for all attributes being passed to the application, not for each attribute. See the documentation for your web browser to determine the cookie size limit.

To configure WS-Federation single sign-on

1. Access the Authentication Scheme Properties dialog for the WS-Federation scheme.
2. Click Additional Configuration.
The WS-Federation Auth Scheme Properties dialog opens.
3. Select the SSO tab.
4. Select a value for the Redirect Mode field.
5. Specify a target resource in the Target field for single sign-on to work. The target specifies the requested resource at the destination Resource Partner and it is required.
6. Optionally, select the Enable Single Use Policy.
7. Click OK to save your configuration.

Create a Custom WS-Federation Authentication Scheme (optional)

The Advanced tab of the Authentication Scheme Properties dialog lets you use a custom WS-Federation scheme written with the SiteMinder Authentication API instead of the existing template provided by SiteMinder.

The Advanced tab contains the Library field. This field contains the name of the shared library that processes WS-Federation authentication. Do not change this value, unless you have a custom authentication scheme, written using the SiteMinder Authentication API.

The default shared library is `smauthsaml`.

Implement WS-Federation Signout

Sign-out is the simultaneous termination of all user sessions for the browser that initiated the sign-out. Closing all user sessions prevents unauthorized users from gaining access to resources at the Resource Partner.

Sign-out does not necessarily end all sessions for a user. For example, a user with two browsers open can have two independent sessions. Only the session for the browser that initiates the sign-out is terminated at all federated sites for that session. The session in the other browser is still active.

The Policy Server performs sign-out using a `signoutconfirmurl.jsp`. This page resides on the Identity Provider system. An Identity Provider initiates a sign-out request on behalf of a user. The JSP sends the sign-out request to each site where the user signed on during a given browser session. The user is then signed out.

A user can initiate a sign-out request only at an Identity Provider. The request is triggered by clicking a link that points to the appropriate servlet. The sign-out confirmation page must be an unprotected resource at the Identity Provider site.

Note: The Policy Server only supports the WS-Federation Passive Request profile for sign-out.

Enable Signout

To configure signout

1. Log on to the FSS Administrative UI.
2. Access the Authentication Scheme Properties dialog for the WS-Federation scheme.
3. Click Additional Configuration.
The WS-Federation Auth Scheme Properties dialog opens.
4. Select the Signout tab.
5. Select the Enable Signout checkbox.
The Signout URL field becomes active.
6. Enter a value for the Signout URL. The URL must begin with `https://` or `http://`.
7. Click OK.
8. Enable the session server.

More Information:

[Storing User Session, Assertion, and Expiry Data](#) (see page 175)

Customize Assertion Processing with the Message Consumer Plug-in

The message consumer plug-in is a Java program that implements the Message Consumer Plug-in. The plug-in lets you implement your own business logic for processing assertions, such as rejecting an assertion and returning a status code. This additional processing works together with the standard processing of an assertion.

Note: For more information about status codes for authentication and disambiguation, see the *SiteMinder Programming Guide for Java*.

During authentication, SiteMinder first tries to process the assertion by mapping a user to its local user store. If SiteMinder cannot find the user, it calls the `postDisambiguateUser` method of the message consumer plug-in.

If the plug-in successfully finds the user, SiteMinder proceeds to the second phase of authentication. If the plug-in cannot map the user to a local user store, the plug-in returns a `UserNotFound` error. The plug-in can optionally use the redirect URL feature. Without the consumer plug-in, the redirect URLs are based on the error that the SAML authentication scheme generates.

During the second phase of authentication, SiteMinder calls the `postAuthenticateUser` method of the message consumer plug-in, if the plug-in is configured. If the method succeeds, SiteMinder redirects the user to the requested resource. If the method fails, you can configure the plug-in to send the user to a failure page. The failure page can be one of the redirect URLs that you can specify with the authentication scheme configuration.

Additional information about the message consumer plug-in can be found as follows:

- Reference information (method signatures, parameters, return values, data types), and the constructor for `UserContext` class, are in the *Java Developer Reference*. Refer to the `MessageConsumerPlugin` interface.
- Overview and conceptual information about authentication and authorization APIs, see the *SiteMinder Programming Guide for Java*.

To configure the plugin

1. Install the SiteMinder SDK, if you have not done so already.
2. Implement the MessageConsumerPlugin.java interface, which is part of the SiteMinder SDK.
3. Deploy your message consumer plug-in implementation class.
4. Enable the message consumer plug-in in the Administrative UI.

Implement the MessageConsumerPlugin Interface

Create a custom message consumer plug-in by implementing the MessageConsumerPlugin.java interface. The minimum requirements for the implementation class are listed in the following procedure.

Follow these steps:

1. Provide a public default constructor method that contains no parameters.
2. Provide code so that the implementation is stateless. Many threads must be able to use a single plug-in class.
3. Implement methods in the interface as your requirements demand.

The MessageConsumerPlugin includes the following four methods:

init()

Performs any initialization procedures that the plug-in requires. 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. SiteMinder calls this method once for each plug-in instance, when SiteMinder is shutting down.

postDisambiguateUser()

Provides processing to disambiguate a user when the authentication scheme is unable to do so. Alternatively, this method can add data for new federation users to a user store. This method receives the decrypted assertion. The decrypted assertion is added to the properties map passed to plug-in under the key "_DecryptedAssertion".

postAuthenticateUser()

Provides additional code to determine the final outcome of assertion processing, regardless of whether the Policy Server processing is a success or failure.

SiteMinder provides the following samples of the Message Consumer plug-in class:

MessageConsumerPluginSample.java in
installation_home\sdk\samples\messageconsumerplugin

MessageConsumerSAML20.java in
installation_home\sdk\samples\authextensionsaml20

Deploy a Message Consumer Plug-in

After you have coded your implementation class for the MessageConsumerPlugin interface, compile it and verify that SiteMinder can find your executable file.

To deploy the Message Consumer Plugin:

1. Compile the MessageConsumerPlugin Java file. The file requires the following dependent libraries, which are installed with the Policy Server:

installation_home\siteminder\bin\jars\SmJavaApi.jar

An identical copy of SmJavaApi.jar is installed with SiteMinder SDK. The file is in the directory *installation_home*\sdk\java\SmJavaApi.jar.

You can use either of them at development time.

2. When a plug-in class is available, in a folder or a jar file, modify the -Djava.class.path value in the JVMOptions.txt file. This step enables the plug-in class to load with the modified classpath. Locate the JVMOptions.txt file in the directory *installation_home*\siteminder\config.

Note: Do not modify the classpath for the existing xerces.jar, xalan.jar, or SmJavaApi.jar.

3. Restart the Policy Server to pick up the latest version of MessageConsumerPlugin. This step is necessary each time the plug-in Java file is recompiled.
4. Enable the plug-in.

Enable the Message Consumer Plug-in for WS-Federation

To customize assertion processing

1. Log on to the FSS Administrative UI.
2. Access the Authentication Scheme Properties dialog for the WS-Federation scheme.
3. Click Additional Configuration.
The WS-Federation Auth Scheme Properties dialog opens.
4. Select the Advanced tab.

5. Complete the following fields:

Full Java Class Name

Specify the Java class name for the plug-in, For example, a sample class included with the SiteMinder SDK is:

`com.ca.messageconsumerplugin.MessageConsumerPluginSample`

Specify a plug-in for each authentication scheme.

Parameter

Specify a string of parameters that are passed to the plug-in specified in the Full Java Class Name field.

As an alternative to configuring the plug-in in the Administrative UI, use the Policy Management API (C or Perl) to set the `IdpPluginClass` and `IdpPluginParameters`.

Redirect Users After Failed Authentication Attempts

For single sign-on processing, you can configure several optional redirect URLs if a user cannot be authenticated at the Resource Partner. The redirect URLs allow finer control over where a user is redirected if the assertion is not valid. For example, if a user cannot be located in a user store, you can fill in a Redirect URL for the User Not Found and send the user to a registration page.

Note: These URLs are not required.

If you do not configure redirect URLs, standard SiteMinder processing takes place. How a failed authentication is handled depends on the configuration.

To configure optional Redirect URLs

1. Access the Authentication Scheme Properties dialog for the WS-Federation scheme.
2. Click Additional Configuration.

The WS-Federation Auth Scheme Properties dialog opens.

3. Select the Advanced tab.
4. Fill in a URL for one or more of the following fields:
 - Redirect URL for the User Not Found status
 - Redirect URL for the invalid SSO Message status
 - Redirect URL for the Unaccepted User Credential (SSO Message) status

If enter a value for the Redirect URL for the Invalid SSO Message status, select a mode.

Note: Click Help for descriptions of settings and controls, including their respective requirements and limits.

Federation Web Services handles the errors by mapping the authentication reason into one of the configured redirect URLs, then the user can be redirected to that URL to report the error.

Note: These redirect URLs can be used with the SiteMinder Message Consumer Plug-in for further assertion processing. If authentication fails, the plug-in can send the user to one of the redirect URLs you specify.

Supply SAML Attributes as HTTP Headers

An assertion response can include attributes in the assertion. These attributes can be supplied as HTTP header variables so a client application can use them for finer grained access control.

The benefits of including attributes in HTTP headers are as follows:

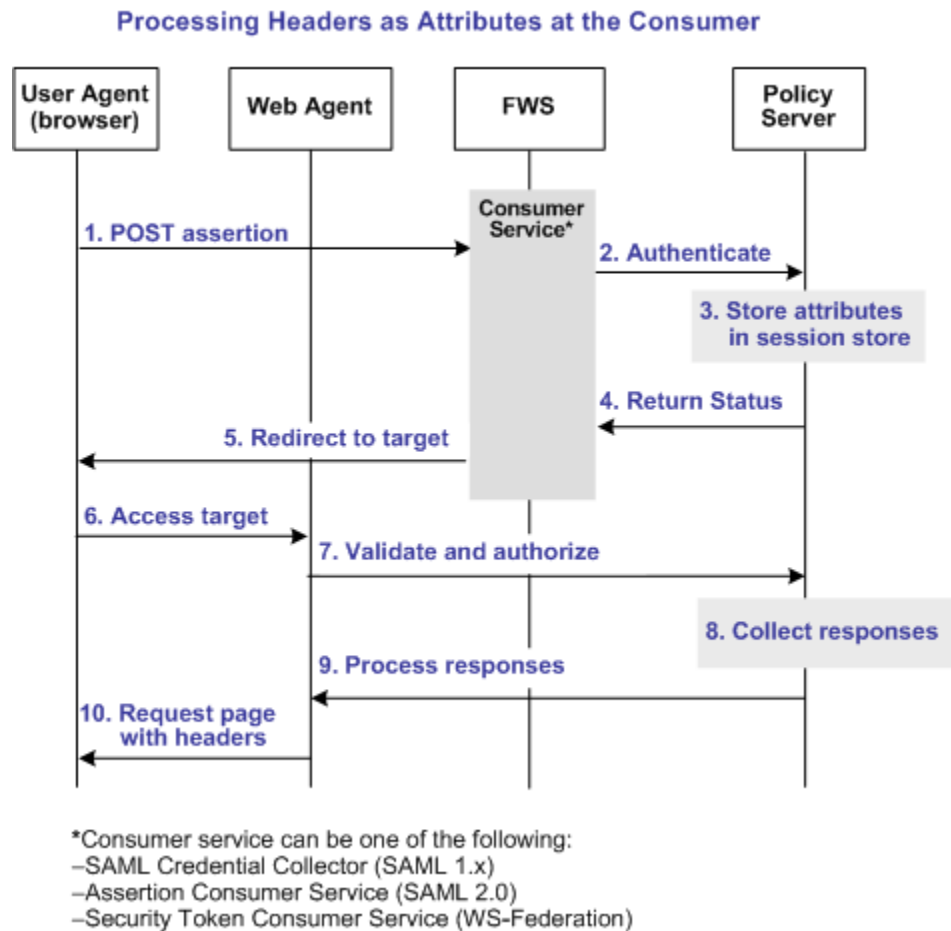
- HTTP headers are not persistent. They are present only within the request or response that contains them.
- HTTP headers, as supplied by the SiteMinder Web Agent, are not visible in the browser, which reduces security concerns.

Note: The HTTP headers have size restrictions that the attributes cannot exceed. SiteMinder can send an attribute in a header up to the web server size limit for a header. Only one assertion attribute per header is allowed. See the documentation for your web server to determine the header size limit.

Use Case for SAML Attributes As HTTP Headers

During authentication, a series of SAML attributes are extracted from an assertion and supplied as HTTP headers. During the authorization process, these headers are returned to the customer application.

The following flow diagram shows the sequence of events at runtime:



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

To process the attributes as HTTP headers, the sequence of events is as follows:

1. After the assertion is generated at the asserting party, it sends the assertion to the appropriate consumer service at the relying party. The delivery mechanism (POST or Artifact or WS-Fed) is irrelevant.

Note: The consumer service can be the SAML credential collector (SAML 1.x), the Assertion Consumer Service (SAML 2.0), or Security Token Consumer Service (WS-Federation).

2. The consumer service calls its local Policy Server to use the configured authentication scheme to authenticate the user with the assertion.

3. If the authentication scheme redirect mode parameter is set to `PersistAttributes`, the Policy Server caches the attributes in the session store as session variables.
4. The result of the authentication is returned to the consumer service.
5. The consumer service redirects the browser to the protected target resource.
6. The browser tries to access the target resource.
7. The Web Agent calls the Policy Server to validate the user session and to verify that the user is authorized to access the target resource.
8. The Policy Server retrieves the attributes by a configured response.
9. The Policy Server processes the responses and sends the attributes to the Web Agent.
10. The Web Agent sets the HTTP headers as necessary.

Configuration Overview to Supply Attributes as HTTP Headers

Several configuration steps are required to retrieve the SAML attributes cached in the session store and provide them as HTTP headers.

Follow these steps:

1. Select `PersistAttributes` as the redirect mode for the SAML authentication scheme, which enables the SAML Attributes to be returned as HTTP headers.
2. Configure an authorization rule for the realm that contains the target resource.
3. Set `PersistentRealm` in the realm protecting the target resource.
4. Configure a response that uses the active response type for each SAML attribute to be supplied as a header.
5. Create a policy that binds the authorization rule and active response to implement the user of attributes as HTTP headers.

Set the Redirect Mode to Store SAML Attributes

After the consuming partner authenticates the user with the SAML assertion, the SAML attributes are written to the session store. The browser is then redirected to the target resource.

To redirect the browser with the attribute data

1. Log in to the FSS Administrative UI.
2. Access the SAML authentication scheme properties dialog.
The properties dialog opens.

3. Set the Redirect Mode parameter to PersistAttributes.

For SAML 1.x, the Redirect Mode is on the Scheme Setup tab. For SAML 2.0 and WS-Federation, the Redirect Mode is on the SSO tab accessed from the authentication scheme properties dialog.

4. Click OK to save your changes.

The redirect mode is now set to pass on the attribute data.

Create an Authorization Rule to Validate Users

For the realm containing the protected target resource, create a rule that is triggered during the authorization process to retrieve the SAML attributes from the session store.

The rule is based on an authorization event (`onAccessAccept`). Because the user has already been authenticated by the FWS application, the Web Agent cannot reauthenticate the user and pass on the HTTP headers. The retrieval of the attributes happen during the authorization stage.

To create an `OnAccessAccept` Rule for the realm

1. Log on to the FSS Administrative UI.
2. From the Domains tab, navigate to the realm which protects the target resource.
3. Select the realm with the target resource and select Create Rule under Realm.
The Rule Properties dialog opens.
4. Enter a name in the Name field that describes the rules purpose as an authorization rule.
5. Select the realm protecting the target resource for the Realm field.
6. Enter an asterisk (*) in the Resource field.
7. Select Authorization events and `OnAccessAccept` in the Action section.
8. Verify that Enabled is selected in the Allow/Deny and Enable/Disable section.
9. Click OK to save the rule.

The authorization rule is now defined for the realm with the protected resource.

Configure a Response to Send Attributes as HTTP Headers

Configure a response that sends the SAML attributes as HTTP headers to the Web Agent. The Web Agent processes the response and makes the header variables available to the client application.

To create a response to send the attributes as headers

1. Log on to the FSS Administrative UI.
2. From the Domains tab, select the domain that contains the target resource.
3. In the domain, select the Responses object and create a response.
The Response Properties dialog opens.
4. Click Create.
The Response Attribute dialog opens.
5. Select WebAgent-HTTP-Header-Variable in the Attribute field.
6. Select Active Response in the Attribute Kind section.
7. Complete the fields in the Attribute Fields section as follows:

Variable Name

Specify the name you want for the header variable. You assign this name.

Library Name

smfedattrresponse

This value must be the entry for this field.

Function Name

getAttributeValue

This value must be the entry for this field.

Parameters

Specify the name of the attribute as it appears in the assertion.

An agreement between you and your federated partner determines the attributes that are in the assertion.

8. Click on OK to save the attribute.
9. Repeat the procedure for each attribute that must become an HTTP header variable. You can configure many attributes for a single response.

The response sends the attributes on to the Web Agent to become HTTP headers.

Create a Policy to Implement Attributes as HTTP Headers

To implement the use of SAML attributes as HTTP headers, you must group together the authorization event rule and active response in a policy.

To create the policy to generate HTTP Headers from SAML attributes

1. Log on to the FSS Administrative UI.
2. From the Domains tab, select the domain that contains the target resource.
3. In the domain, select the Policies object and create a policy.
The Policy Properties dialog opens.
4. Enter a descriptive name in the Name field.
5. Select the users that must have access to the protected resource in the Users tab.
6. Add the authorization rule you created previously on the Rules tab.
7. Select the authorization rule and click Set Response.
The Available Responses dialog opens.
8. Select the active response you created previously and click OK.
You return to the Rules tab. The response appears with the authentication rule.
9. Click OK to save the policy.

The policy that enables SAML attributes to be used as HTTP headers is complete.

How To Protect a Target Resource with a WS-Federation Authentication Scheme

Protect target federation resources by configuring a SiteMinder policy that uses the WS-Federation authentication scheme.

Follow these steps:

1. Create a realm that uses the WS-Federation authentication scheme. The realm is the collection of target resources.

You can create a realm in the following ways:

- Create a unique realm for each authentication scheme already configured.
- [Configure a single target realm](#) (see page 279) that uses a custom authentication scheme to dispatch requests to the corresponding WS-Federation authentication schemes. Configuring one realm with a single target for all producers simplifies configuration of realms for authentication.

2. Configure an associated rule and optionally, a response.
3. Group the realm, rule, and response into a policy that protects the target resource.

Important! Each target URL in the realm is also identified in an unsolicited response URL. An unsolicited response is sent from the Account Partner to the Resource Partner, without an initial request from the Resource Partner. In this response is the target. At the Account Partner, an administrator includes this response in a link. The link redirects the user to the Resource Partner.

Configure a Unique Realm for Each WS-Fed Authentication Scheme

The procedure for configuring a unique realm for each WS-Federation authentication scheme (artifact or profile) follows the standard instructions for creating realms in the FSS Administrative UI.

To create a realm for each WS-Federation authentication scheme

1. Log on to the FSS Administrative UI.
2. Click the System tab.
3. Click Edit, System Configuration, Create Domain.
The Domain dialog opens.
4. Create a policy domain.
5. Create a realm under the policy domain from the previous step, noting the following:
 - a. Select the Web Agent protecting the web server where the target federation resources reside for the Agent field.
 - b. Select the WS-Federation authentication scheme for the Authentication Scheme field. This authentication scheme protects the realm.
6. Create a rule for the realm.
As part of the rule you select a Web Agent action (Get, Post, or Put), which allows you to control processing when users authenticate to gain access to a resource.
7. Configure the policy, using the realm you created.
8. Save the policy.
9. Exit the FSS Administrative UI.

Note: Click Help for descriptions of settings and controls, including their respective requirements and limits.

A policy with a unique realm now protects the federated resource.

Configure a Single Target Realm for All WS-Federation Authentication Schemes

To simplify configuration of realms for all WS-Federation authentication schemes, create a single target realm for multiple Account Partners.

To do this task, set-up:

- A single custom authentication scheme
You must configure a WS-Federation authentication scheme for each Account Partner before configuring a custom template.
- A single realm with one target URL

Configure WS-Federation Authentication Schemes for the Single Target Realm

Configure the necessary WS-Federation authentication schemes that are referenced by the custom authentication scheme you associate with the single target realm. When you define the custom authentication scheme, you define a parameter that instructs the Policy Server which authentication schemes that the custom scheme uses.

To create the WS-Federation authentication scheme

1. Log on to the FSS Administrative UI.
2. Create the WS-Federation authentication schemes according to the procedures in this guide.
3. Exit the FSS Administrative UI.

Configure a Custom WS-Federation Auth. Scheme

A single target realm relies on a custom authentication scheme to work.

To configure a custom authentication scheme for a single target realm

1. Log on to the FSS Administrative UI.
2. Select the System tab.
3. Select Edit, System Configuration, Create Authentication Scheme.

The Authentication Scheme Properties dialog opens.

4. Complete the fields as follows:

Note: Click Help for descriptions of settings and controls, including their respective requirements and limits.

Name

Enter a descriptive name to indicate the custom auth scheme, such as WS-Fed Custom Auth Scheme.

5. Complete the following field in the Scheme Common Setup section:

Authentication Scheme Type

Custom Template

6. Complete the following fields in the Scheme Setup tab

Library

smauthsinglefed

Secret and Confirm Secret

Leave this field blank.

Confirm Secret

Leave this field blank

Parameter

Instructs the custom scheme which WS-Federation authentication schemes it must use. Specify one of the following options:

- SCHEMESET=LIST;Scheme1;Scheme2;
Specifies list of target WS-Federation authentication scheme names to use (Scheme1 and Scheme2 are examples)
- SCHEMESET=WSFED_PASSIVE;
The smauthsinglefed scheme enumerates all WS-Federation authentication schemes to find the one with correct Provider Source Id.

Enable this scheme for SiteMinder Administrators

Leave unchecked.

7. Click OK to save your changes.

Configure the Single Target Realm

After you configure the authentication schemes, including the custom authentication scheme, you can configure a single target realm for federation resources.

To create the single target realm

1. Log in to the FSS Administrative UI.
2. Select the Domains tab.
3. Select the policy domain you previously created for the single target realm.
4. Select the Realms object and select Edit, Create Realm.

The Realm Properties dialog opens.

5. Enter the following values to create the single target realm:

Note: Click Help for descriptions of settings and controls, including their respective requirements and limits.

Name

Enter a name for this single target realm.

6. Complete the following field in the Resource section:

Agent

Select the SiteMinder Web Agent protecting the web server with the target resources.

Resource Filter

Specify the location of the target resources. Any user requesting a federated resource is be redirected to this location.

For example, /FederatedResources.

7. Select the Protected option in the Default Resource Protection section.
8. Select the previously configured custom authentication scheme in the Authentication Scheme section. This custom authentication uses the smauthsinglefed library.

For example, if the custom scheme was named Fed Custom Auth Scheme, you must select this scheme.
9. Click OK.

The single target realm task is complete.

Create a Policy Using the Single Target Realm

Create a policy that references the single target realm. Remember that the single target realm uses the custom authentication scheme that directs requests to the appropriate SAML/WS-Fed authentication scheme.

Note: This procedure assumes that you have already configured the domain, custom authentication scheme, single target realm and associated rule.

To create a policy and add it to an existing domain

1. Log on to the FSS Administrative UI.
2. From the Domains tab, select the Policies object.
3. Click Edit, Create Policy.

The Policy Properties dialog opens.
4. Enter a name and a description of the policy in the General section.

5. Add users to the policy from the Users tab.
6. Add the rule you created for the single target realm from the Rules tab.
The remaining tabs are optional.
7. Click OK.

The policy task is complete.

Chapter 20: Use SAML 2.0 Provider Metadata To Simplify Configuration

This section contains the following topics:

[SiteMinder SAML 2.0 Metadata Tools Overview](#) (see page 449)

[Export Metadata Tool](#) (see page 450)

[Import Metadata Tool](#) (see page 457)

SiteMinder SAML 2.0 Metadata Tools Overview

SiteMinder provides a metadata tool to import and export SAML 2.0 metadata programmatically. Metadata lets you efficiently exchange federation configurations between a site that uses SiteMinder and a partner that uses a third party or SiteMinder. Programmatic use of SAML 2.0 metadata can limit how much configuration that you perform.

The Policy Server installs the metadata tool. The two command-line utilities that make up the SiteMinder metadata tools are smfedexport and smfedimport

Exporting metadata involves the following types of input:

- User input
- Access to the SiteMinder smkeydatabase for including KeyInfo into the metadata
- Access to the smkeydatabase for signing
- Access to the policy store to reference similar metadata that can be used as a template.

Importing metadata involves:

- User input
- Access to the policy store
- Access to the smkeydatabase for verifying signatures, if certificates are configured
- Parsing the XML metadata in the metadata document
- Storing the relevant metadata in the policy store
- Storing the PKI information from the metadata in the smkeydatabase

Export Metadata Tool

You can use the export tool in the following situations:

- Create an Identity Provider metadata file for use by Service Providers.

Use the tool to produce a metadata file containing information about profiles that the Identity Provider supports. This XML output that the export tool generates describes the Identity Provider. Sites acting as Service Providers can import this metadata file to establish a relationship with the Identity Provider.

- Create an Identity Provider metadata file from an existing Service Provider.

A SiteMinder Identity Provider generates a metadata file from an existing Service Provider object. The use of the Service Provider object reduces the amount of required data that a user must configure. Many of the settings for the Identity Provider metadata file can be derived from the existing Service Provider. Also, SiteMinder provides the default names of the servlets.

To use the metadata file, the existing relationship between the Identity Provider and the Service Provider is similar to the relationship you are establishing.

The SSO and SLO servlet URLs are the default servlet names that are prepended with the IP address and port of the Federation Web Services application.

The servlet names are:

- `http://idp_server:port/affwebservices/public/saml2sso`
- `http://idp_server:port/affwebservices/public/saml2slo`

idp_server:port

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

- Create a Service Provider metadata file for Use by Identity Providers.

A SiteMinder Service Provider can facilitate federation with sites acting as Identity Providers by producing a metadata file containing information about the profiles it supports. An Identity Provider can import the metadata file to establish a relationship with the Service Provider.

- Create a Service Provider metadata file from an existing SAML 2.0 Authentication Scheme.

A SiteMinder Service Provider generates a metadata file from an existing SAML 2.0 Authentication Scheme object. The use of the Service Provider object reduces the amount of required data that a user must configure. Many of the settings for the SP metadata file can be derived from the existing SAML 2.0 authentication scheme. SiteMinder provides the default names of the servlets.

To use the metadata file, the existing relationship between the Service Provider and the Identity Provider must be similar to the relationship you are establishing. The SSO and SLO servlet URLs are the default servlet names that are prepended with the IP address and port of the Federation Web Services application.

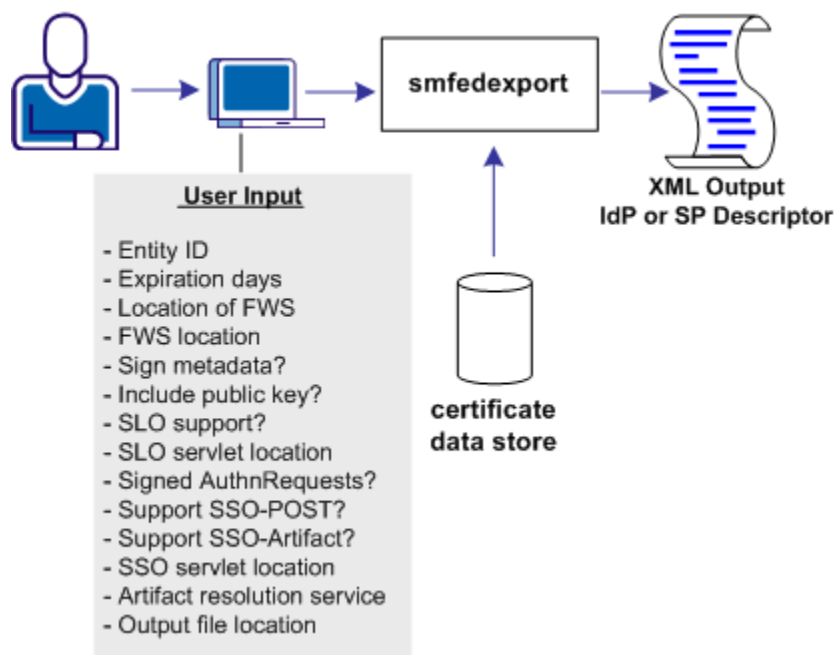
The servlets are:

- `http://idp_server:port/affwebservices/public/saml2sso`
- `http://idp_server:port/affwebservices/public/saml2slo`

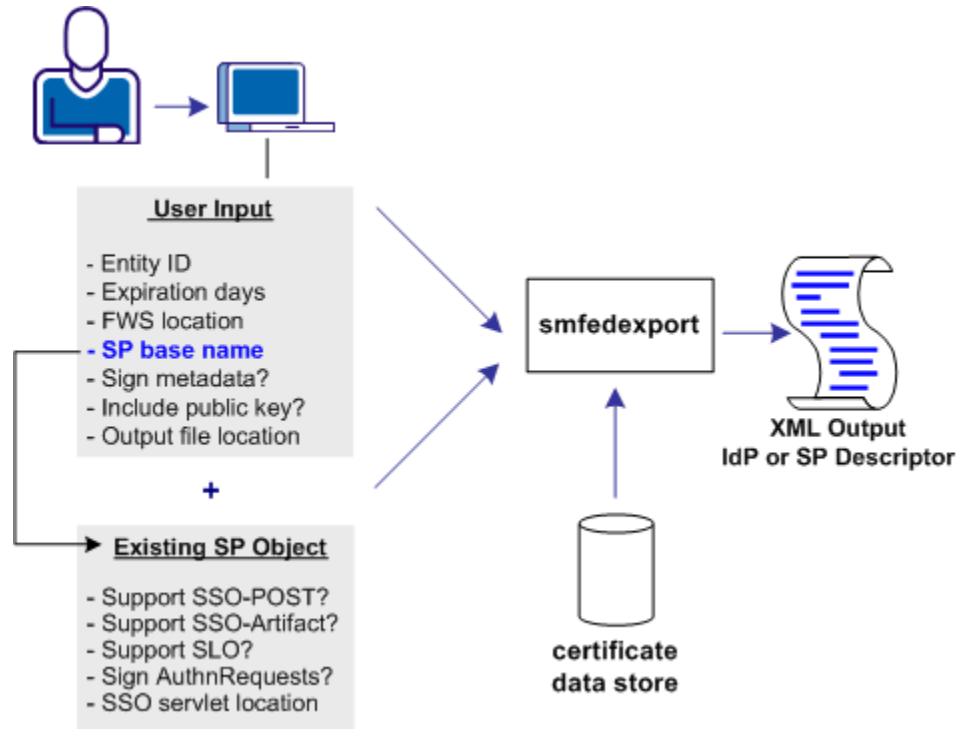
idp_server:port

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

The following illustration shows a metadata file that is generated only from user input.



The following illustration shows a metadata file that is generated from a combination of user input and data from an existing Service Provider object.



Run the smfedexport Tool

The smfedexport tool lets you export SAML 2.0 metadata to an XML file.

If you enter smfedexport without any command arguments, all the command arguments and their usage are displayed.

To run the smfedexport tool

1. At the system where you installed the Policy Server, open a command window.
2. Enter the smfedexport command using the syntax for the task you want to complete:

Note: Command arguments enclosed in square brackets [] are optional.

To export a SAML 2.0 Identity Provider metadata file:

```

smfedexport -type saml2idp [-entityid <entityid>] [-expiredays <num>]
[-fwsurl <FWS Location>] [-spbase <spname>] -username <SiteMinder Admin Name>
-password <SiteMinder Admin Password>][-sign][pubkey]
[-slo <SLO Service Location> -slobinding <REDIR>] [-reqsignauthr]
[-sso <SSO Service Location> -ssobinding <REDIR|SOAP>]
[-ars <Artifact Resolution Service Location>][output <file>]
  
```

To export a SAML 2.0 Service Provider metadata file:

```
smfedexport -type saml2sp [-entityid <entityid>] [-expiredays <num>]
[-fwsurl <FWS Location> [-schemebase <Auth Scheme name>
-username <SiteMinder Admin Name> -password <SiteMinder Admin Password>]]
[-sign][[-pubkey][[-slo <SLO Service Location> -slobinding <REDIR>]
[-signauthr][[-acs <Assertion Consumer Service> -acsbinding <ART|POST|PAOS>
-acsindex <num>][[-acsisdef]]][-output <file>]
```

To sign an existing Metadata document:

```
smfedexport -type (saml2sp|saml2idp) -sign -input <file> -output <file>
```

After you run the tool, an XML file will be produced. If the `-type` option is set to `saml2idp`, the default output file name is `IDPSSODescriptor.xml`. If the `-type` option is set to `saml2sp`, the default output file name is `SPSSODescriptor.xml`.

After `smfedexport` processes the initial command options, the tool prompts you for additional data that is related to the type of export file the tool is generating. Any optional arguments that you do not enter use default values.

Note: If you are creating an IdP metadata file, you must have at least one single sign-on service defined in the `smfedexport` command. If you are creating an SP metadata file, you must have at least one assertion consumer service defined in the `smfedexport` command.

Command Options for smfedexport

The `smfedexport` command line options are listed in the table that follows:

Option	Description	Values
<code>-acs</code>	Assertion Consumer Service URL	URL
<code>-acsindex</code>	Assertion Consumer Service index value	integer
<code>-acsisdef</code>	Makes the immediately preceding Assertion Consumer Service the default.	none
<code>-acsbinding</code>	SAML protocol binding for the Assertion Consumer Service.	<ul style="list-style-type: none"> ■ ART (for artifact) ■ POST (for POST) ■ PAOS (for Reverse SOAP - ECP)
<code>-ars</code>	Artifact Resolution Service	URL
<code>-entityid</code>	Represents the ID of the SP or IDP whose metadata you are exporting	URI

Option	Description	Values
-expiredays	Days until the metadata document is no longer valid	integer, 0 is the default A value of 0 indicates that the metadata document has no expiration and results in no "validUntil" elements being generated in the exported XML
-fwsurl	URL pointing to the FWS application.	URL in the form <i>http://host:port</i>
-input	Full path to an existing XML file	string, no default
-output	Full path to an output XML file	Default values: IDPSSODescriptor.xml SPSSODescriptor.xml
-password	SiteMinder Administrator name Requires the -username option	string, no default
-pubkey	Tells the Policy Server to include the certificate (public key) in the metadata. The partner site uses the public key for signature encryption and verification. This setting is optional because the metadata must not be signed.	true, if present false otherwise
-reqsignauthr	Require signed AuthnRequests	true, if present false otherwise
-schemebase	Points to an existing Service Provider. The settings for the profiles/bindings are taken from this provider. Requires the following options: -fwsurl -username -password	authentication scheme name
-spbase	Points to an existing Service Provider. The settings for the profiles/bindings are taken from this provider. Requires the following options: -fwsurl -username -password	Service Provider Name
-sign	Indicates whether the Policy Server signs the metadata. This setting is optional.	true, if present false, otherwise

Option	Description	Values
-sigalg	Designates the signature hashing algorithm SiteMinder uses to for signing assertions and assertion responses, single logout requests and responses	rsawithsha1 rsawithsha256
-signauthr	Indicates whether the SP signs AuthnRequests	true, if present false, otherwise
-signingcertalias	Specifies the alias associated with the key/certificate pair that signs the metadata. The pair must be stored in the smkeydatabase. This setting is an alternative to the default alias, defaultenterpriseprivatekey. If you do not enter a value for this option, the Policy Server uses the defaultenterpriseprivatekey alias to sign the metadata.	alias name
-slo	Single Logout Service URL	URL
-slobinding	HTTP binding used for single logout. HTTP Redirect binding is the only option.	
-sso	Single sign-on service URL	URL
-ssobinding	SSO Service URL protocol binding	<ul style="list-style-type: none"> ■ REDIR (for web SSO) ■ SOAP (for ECP)
-type (Required)	Entity type of the export file	saml2idp sam2sp
-username	The SiteMinder Administrator name, which requires the -password option.	string, no default

smfedexport Tool Examples

Example: Exporting an Identity Provider

```
smfedexport -type saml2idp -entityid http://www.myidp.com/idp1
-expiredays 30 -sign -pubkey -slohttpredir http://www.mysite.com
/affwebservices/public/saml2slo -reqsignauthr
-ssoart http://www.mysite.com/affwebservices/public/saml2sso
-artressvc http://www.mysite.com/affwebservices/
saml2artifactresolution -output myidpdescription.xml
```

Example: Exporting a Service Provider

```
smfedexport -type saml2sp -entityid http://www.myidp.com/sp1
-expiredays 30 -sign -pubkey -slohttpredir http://www.mysite.com/
affwebservice/public/saml2slo -signauthr -aconsvcpost
http://www.mysite.com/affwebservice/public/saml2assertionconsumer
-aconsvcpostindex 12345 -output myidpdescription.xml
```

Example: Modifying and Signing an Exported Data File

In this example, you are modifying and digitally signing an XML file using the smfedexport.

To modify and sign a metadata file

1. Edit the existing XML file using an XML editor.
2. Enter the following command:

```
smfedexport -sign -infile file -output file
```

For example:

```
smfedexport -sign -infile myspdescription.xml -output newspdescription.xml
```

To modify an exported file that is already digitally signed

1. Edit the existing XML file using an XML editor as need.
2. Delete the <Signature> element from the file.
3. Enter the following command:

```
smfedexport -sign -infile file -output file
```

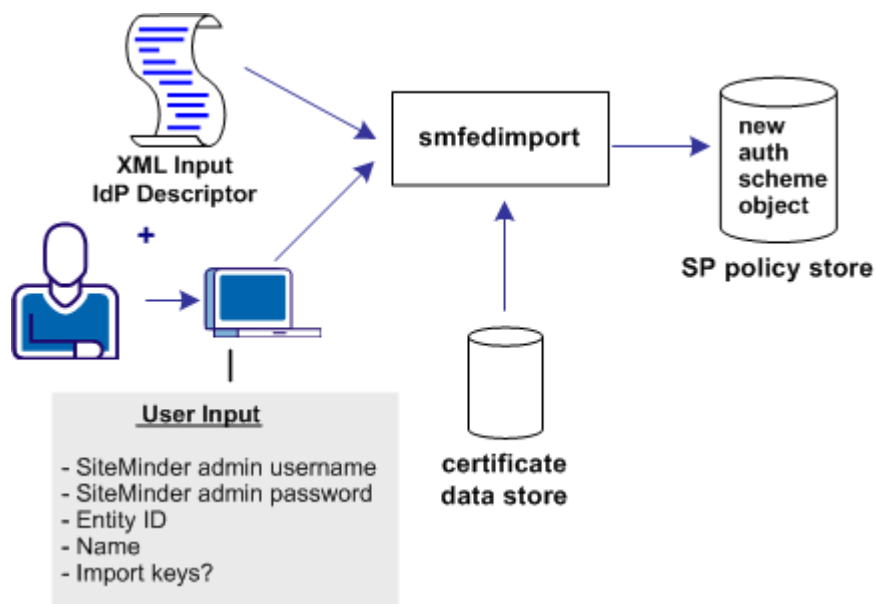
For example:

```
smfedexport -sign -infile myspdescription.xml -output newspdescription.xml
```

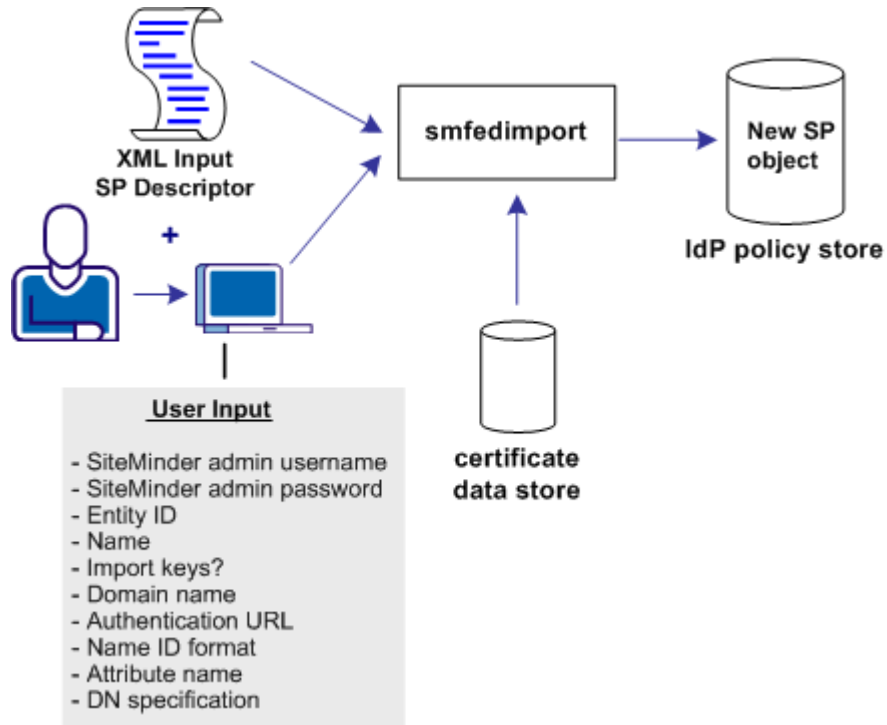

Import Metadata Tool

You can use the import tool for the following tasks:

- Create a SAML 2.0 authentication scheme for a Service Provider, as shown in the following illustration.



- Create a SAML 2.0 Service Provider object for an Identity Provider.



Run the smfedimport Tool

The smfedimport utility can import SAML Identity Providers and Service Providers into a SiteMinder policy store and smkeydatabase. If you import a Service Provider input file, the result is a new SiteMinder Service Provider object within an existing affiliate domain. If you import a SAML Identity Provider input file, the result is an authentication scheme based on the SiteMinder SAML 2.0 Template.

When the smfedimport command line utility is run, the first and second parameters are the username and password of the SiteMinder administrator. The third and final argument is the path to the input XML file.

To run the smfedimport tool

1. At the system where you installed the Policy Server, open up a command window.
2. Enter the command using the following syntax:

To import a SAML2 Identity Provider metadata file into the policy store:

```
smfedimport -type saml2idp -username <username>
-password <password> -entityid <entityid> -name <name>
[-importkeys <name>] [-silent] -input <file>
```

To import a Service Provider metadata file into the policy store:

```
smfedimport -type saml2sp -username <username>
-password <password> -entityid <entityid> -domainname <name>
-authurl <URL> -nameidformat (U|E|X|W|K|N|P|T|U)
-nameidtype (S | U | D) -attrname <name> -dnspec <spec>
-name <name>[-importkeys <name>] [-silent] -input <file>
```

Note: Switches in square brackets [] are optional.

After smfedimport processes the initial command options, the tool prompts you for additional data based on the type of file you are importing. Any optional arguments that you do not enter on the command line have default values.

smfedimport Tool Examples

Example: Importing Identity Provider metadata

```
smfedimport -type saml2idp -username Siteminder
-password siteminderpassword -entityid http://www.myidp.com
-name mynewauthscheme -importkeys keyaliasname -input mypartnersidpinfo.xml
```

Example: Importing Service Provider metadata

```
smfedimport -type saml2sp -username Siteminder -password siteminderpassword
-entityid http://www.mysp.com -name mynewsaml2sp -importkeys
keyalisname -domainname myaffiliatedomain
-authurl http://www.mysite.com/login.html -nameidformat U
-nameidtype S -attrname attrname -input mypartnersspinfo.xml
```

Command Options for smfedimport

The command line options are listed in the following table.

Option	Description	Value
-attrname	Attribute name required for nameID	string
-authurl	Authentication URL	URL
-dnspec	DN specification required for name ID type only	string
-domainname	Affiliate domain name	string

Option	Description	Value
-entityid	Entity ID	The Service Provider ID for the import or the Identity Provider ID for the import
-importkeys	Indicates whether the certificates in the metadata are imported into smkeydatabase.	string. Enter a name that becomes an alias associated with the certificate in smkeydatabase. If there are multiple certificates, the aliases are added as name, name1, name2.
-input	input file	string
-name	Indicates the name of the SiteMinder object, such as the name of the Service Provider or the name of a SAML authentication scheme	string
-nameidformat	Name ID format	(U)nspecified--default (E)mail address (X)509 Subject name (W)indows domain name (K)erberos Principal Name E(n)tity Identifier (P)ersistent Identifier (T)ransient Identifier
-nameidtype	Name ID type	(S)tatic (U)ser attribute (D)N attribute
-password	SiteMinder Administrator password	string, no default
-type (Required)	Entity type of the import file	saml2idp sam2sp
-silent	Determines whether the tool interactively prompts the user	true, if present false otherwise
	With this option, the tool operates in silent mode. The tool does not interactively prompt the user for missing input. The tool also does not prompt the user to accept the import of each separate entity in the input file. The tool assumes that all entities in the input file must be imported.	
-username	SiteMinder Administrator name	string, no default

Processing Import Files with Multiple SAML 2.0 Providers

If multiple providers are specified in one import file, the tool imports them into the same affiliate domain. The names for each provider are based on the value you specify for the `smfedimport` command option **-name**.

For example, if there are three Service Providers in the import file and you specify:

```
-name mySP
```

The tool registers the imported providers as `mypsp`, `mypsp_1`, and `mypsp_2`. The integer increases by one for each subsequent provider. If there is a mixture of Identity Providers and Service Providers in an import file, the naming convention still applies.

Processing Import Files with Multiple Certificate Aliases

If there are multiple certificates in the import file, the tool imports them into the `smkeydatabase`. The tool then assigns alias names based on the value you specify with `smfedimport` command option **-importkeys**.

For example, if there are three certificates in the import file and you specify:

```
-importkeys myalias
```

The tool registers the imported certificates as `myalias`, `myalias_1`, and `myalias_2`. The integer increases by one for each subsequent certificate.

Chapter 21: Federation Security Services

Trace Logging

This section contains the following topics:

- [Trace Logging](#) (see page 463)
- [FWS Log Messages at the Web Agent](#) (see page 463)
- [FWS Log Messages at the Policy Server](#) (see page 465)
- [Update Federation Web Services Data in the Logs](#) (see page 467)
- [Simplify Logging with Trace Configuration Templates](#) (see page 467)
- [Flush Federation Web Services Cache for Trace Logs](#) (see page 470)

Trace Logging

The Web Agent trace logging facility and the Policy Server Profiler enable SiteMinder to monitor the performance of the Web Agent and Policy Server. These logging mechanisms provide comprehensive information about the operation of SiteMinder processes so you can analyze performance and troubleshoot issues.

For Federation Security Services, several logging components are available to collect trace messages related to federated communication. Trace messages provide detailed information about program operation for tracing, debugging, or both. Trace messages are ordinarily turned off during normal operation. You can enable them to extract in-depth information in addition to the trace message itself. For example, you can look at the FWSTrace.log to see the SAML assertion generated by SiteMinder or collect the name of the current user.

The collected trace messages are written to a trace log. The FWSTrace.log is located in the directory *web_agent_home/log*.

Note: For Web Agents on IIS 6.0 servers, log files are created only after the first user request has been submitted. To verify your configuration in the log file, a user has to submit a request.

FWS Log Messages at the Web Agent

The Federation Web Services (FWS) application that is installed with the Web Agent Option Pack, represents the federation client. The component that controls the trace messages and monitors FWS activity is the Fed_Client component.

Within the Fed_Client component, the following sub components are included:

single sign-on

Monitors single sign-on activity.

single logout

Monitors requests for single logout.

discovery profile

Monitors the identity provider discovery profile activity.

administration

Watches administration-related messages.

request

Monitors request and authentication activity.

general

Monitors activity that other subcomponents are not monitoring.

configuration

Monitors SAML 2.0 Service Provider configuration messages.

FWS uses the common tracing facility that the Web Agent uses to log trace messages. The following files are used to set up trace logging:

trace configuration file

Specifies the configuration file that determines which components and events FWS monitors. The default file is FWSTrace.conf.

trace log file

Specifies the output file for all the logged messages. You provide a name and the location for this file in the Web Agent configuration file.

Web Agent Configuration File or Agent Configuration Object

Contains the logging parameters that enable logging and format the log. This file does not define message content.

Configure FWS Trace Logging

To collect trace messages for the Federation Web Services application, configure the FWS trace logging.

Follow these steps:

1. Do one of the following tasks:
 - Make a copy of the default template, FWSTrace.conf and modify the file to include only the data you want to monitor.
 - Copy one of the preconfigured templates and assign a new name to it.

Note: Do not edit the template directly.
2. Open the LoggerConfig.properties file in the directory *web_agent_home/affwebservices/WEB-INF/classes*, and set the following parameters:
 - Set TracingOn to Yes. This option instructs the trace facility to write messages to a file.
 - Set the TraceFileName parameter to the full path of the trace log file. The default location is *web_agent_home/config/FWSTrace.log*.
 - Set the TraceConfigFile parameter to the full path of the trace configuration file, either the default template, FWSTrace.conf or another template. Templates can be found at *web_agent_home/config*.
3. Optionally, you can format the trace log file, the file that contains the log output. The following parameters are the Web Agent configuration parameters that dictate the format of the trace log file:
 - TraceRollover
 - TraceSize
 - TraceCount
 - TraceFormat
 - TraceDelim

The LoggerConfig.properties file contains descriptions of all these settings.

FWS Log Messages at the Policy Server

The component that controls the trace messages for federation services at the Policy Server is the Fed_Server component. This component monitors activity for the assertion generator and the SAML authentication scheme. For example, you can view the generated assertion in the smtracedefault.log file.

To configure logging at the Policy Server, use the Policy Server Profiler. The Profiler is available from the Policy Server Management Console. The Profiler is a graphical user interface that lets you specify components for trace logging, which include:

trace configuration file

Defines the components and subcomponents that are included in the file.

trace log file

Specifies the output file for all the logged messages.

The following subcomponents are available for the Fed_Server component:

Configuration

Monitors SAML 2.0 Service Provider configuration activity.

Assertion_Generator

Watches the activity for the SAML 1.x and 2.0 assertion generators.

Auth_Scheme

Monitors the activity of the SAML 1.x or SAML 2.0 authentication schemes.

Saml_Requester

Watches SAML Requester activity

Attribute_Service

Watches the Attribute Service activity

Use the SiteMinder Profiler to Log Trace Messages

The profiler is the Policy Server facility for logging. You can use the profiler to collect trace messages for federation services.

Access the profiler from the Policy Server Management Console.

To configure the profiler

1. Open the Policy Server Management Console.
2. Select the Profiler tab.
3. Select the Enable Profiling check box.
4. In the Configuration File field, click Browse and locate the template that you want to use.

You can load the default template, *smtracedefault.txt*, in the directory *policy_server_home/config*, or one of the preconfigured templates in the directory *policy_server_home/config/profiler_templates*.

5. In the Output section, select whether to log data to the Console or to a File or both. If you select a file, specify a path to that file in the Output to File field then select an output format.

Note: Verify that the log file uses a unique name.

6. Click OK to save your changes.

Update Federation Web Services Data in the Logs

If you modify any part of the federation configuration at the producer/Identity Provider or the consumer/Service Provider, flush the Federation Web Services cache for the changes to appear in the trace logs.

Note: Notice a brief delay from when the changes are made and when Federation Web Services receives the information.

To flush the cache

1. Access the FSS Administrative UI.
2. Select Tools, Manage Cache to access the Cache Management dialog.
3. Click Flush All.
4. Click OK.

Simplify Logging with Trace Configuration Templates

To make the task of collecting tracing data simpler, a series of preconfigured templates are installed with the Policy Server and the Web Agent Option Pack. You can use these templates instead of creating your own trace configuration file to collect the data that gets written to a trace log.

Trace Logging Templates for FWS

The following templates are available for Federation Web Services:

Template	Tracing Messages Collected
WebAgentTrace.conf	Default template. Collects data that you specify.
FWS_SSOTrace.conf	Collects single sign-on messages
FWS_SLOTrace.conf	Collects single logout messages

Template	Tracing Messages Collected
FWS_IPDTrace.conf	Collects Identity Provider Discovery Profile messages

All these templates include the Fed_Client component and subcomponents for the specific data being tracked. Look at each template to see the exact contents. The templates are located in *web_agent_home/config*.

To use a template for trace logging

1. Make a copy of the template you want to use and rename the copy.
Note: Do not edit the template directly.
2. Open the Agent configuration file or Agent configuration Object.
3. Set the TraceFile parameter to Yes.
4. Set the TraceFileName parameter to the full path to the trace log file. This file contains the log output.
5. Set the TraceConfigFile parameter to the full path to the newly named template file.
6. Format the trace log file. The following parameters are the Web Agent configuration parameters that dictate the format of the trace log file:
 - TraceAppend
 - TraceFormat
 - TraceDelimiter
 - TraceFileSize
 - LogLocalTime

For descriptions of each logging parameter, see the *Web Agent Configuration Guide*.

Note: Web Agents on IIS 6.0 and Apache 2.0 servers do not support dynamic configuration of log parameters that are set locally in the Agent configuration file. Consequently, when you modify a parameter, the change takes effect only after the Agent is restarted. If you configure the log parameters in an Agent configuration object, these log settings can be stored and updated dynamically.

FWS Template Sample

The following text is an excerpt from the FWS_SLOTTrace.conf template. Most of the file contains comments and instructions on how to use the file, the command syntax, and the available subcomponents for the Fed_Client component.

The excerpt shows the component, Fed_Client and the subcomponents (Single_Logout and Configuration) that are monitored. The excerpt also shows the specific data fields that indicate the required contents of each message (Date, Time, Pid, Tid, TransactionId, SrcFile, Function, Message).

```
components: Fed_Client/Single_Logout, Fed_Client/Configuration
data: Date, Time, Pid, Tid, TransactionID, SrcFile, Function, Message
```

Trace Logging Templates for the IdP and SP

The following templates are available for trace logging related to the Identity Provider and the Service Provider, such as assertion generation or SAML authentication.

Template	Tracing Messages Collected
samlidp_trace.template	Collects messages for Identity Provider activity
samlsp_trace.template	Collects messages for Service Provider activity

Look at each template to see the exact contents. The templates are located in *policy_server_home/config/profiler_templates*.

To use the template

1. Open the Policy Server Management Console.
2. Select the Profiler tab.
3. Select the Enable Profiling check box.
4. In the Configuration File field, click Browse and locate the template that you want to use.
5. In the Output section, select whether to log the data to the Console or to a File or both. If you select a file, specify a path to that file in the Output to File field and select an output format.

Note: Verify that the log file uses a unique name.

6. Click OK to save your changes.

Service Provider Template Sample

The following text is the samlsp_trace.template file.

```
components: Server/Policy_Server_General, IsProtected/Resource_Protection,
Login_Logout/Authentication, Login_Logout/Policy_Evaluation,
Login_Logout/Active_Expression, Login_Logout/Session_Management,
IsAuthorized/Policy_Evaluation, JavaAPI, Fed_Server/Auth_Scheme,
Fed_Server/Configuration
data: Date, Time, Tid, TransactionID, SrcFile, Function, Domain, Resource, Action,
User, Message
```

For Federation Security Services, it includes the Fed_Server component along with the subcomponents Auth_Scheme and Configuration.

The data fields that indicate the required contents of each message are:

Date, Time, Tid, TransactionId, SrcFile, Function, Domain, Resource, Action User, and Message.

Identity Provider Profiler Sample

At the Identity Provider, the Profiler tab of the Policy Server Management Console specifies a template in the Configuration File field. For example, a sample entry for the Configuration File field would be:

```
c:\program files\ca\siteminder\config\profile_templates\samlidp_template.trace
```

For more information about using the Profiler, see the *Policy Server Administration Guide*.

Flush Federation Web Services Cache for Trace Logs

If you modify any part of the federation configuration at the asserting or relying party, flush the Federation Web Services cache for the changes to appear in the trace logs.

To flush the cache

1. Log on to the FSS Administrative UI.
2. Select Tools, Manage Cache
The Cache Management dialog opens.
3. Click Flush All in the All Caches section.
4. Click OK.

All caches is now cleared.

Chapter 22: Configuration Settings that Must Use the Same Values

This section contains the following topics:

[How to Use the Configuration Settings Tables](#) (see page 471)

[SAML 1.x Matching Configuration Settings](#) (see page 471)

[SAML 2.0 Matching Configuration Settings](#) (see page 473)

[WS-Federation Configuration Settings](#) (see page 474)

How to Use the Configuration Settings Tables

When configuring a federated environment, there are many instances where you must configure matching parameter values at both sides of a transaction.

The tables that follow explicitly describe each matching set of parameters. Each cell in a row describes a setting that must match the corresponding value or values described in the other cells in the row.

Note: The information is only applicable in an environment where the asserting and relying party are SiteMinder systems.

SAML 1.x Matching Configuration Settings

The following table lists SiteMinder configuration settings that must be set to the same value at the SAML 1.x producer and consumer. The table also indicates the dialog or file where these settings are located. Most of these settings are in the FSS Administrative UI; however, some parameters are in a properties file or part of a link.

Important! If you have to enter a URL as a value for a setting, the URL string that comes after the colon, for example, "http:" is case sensitive. Therefore, the case of the URLs in all Audience-related settings and Assertion Consumer URL-related settings must match.

These Settings at the SAML 1.x Consumer...	Must Match These Settings at the SAML 1.x Producer...
<p><AffiliateName> AffiliateConfig.xml file for SAML Affiliate Agent OR Affiliate Name field Scheme Setup tab of the Authentication Scheme Properties dialog (Artifact and POST profiles)</p>	<p>Name field Affiliate Properties dialog; value must be lowercase NAME query parameter in intersite transfer URL links at the producer.</p>
<p>Verify Password field (SAML Artifact auth. scheme only) Scheme Setup tab of the Authentication Scheme Properties dialog</p>	<p>Confirm Password field Affiliate Properties dialog</p>
<p><AssertionAudience> setting AffiliateConfig.xml file; SAML Affiliate Agent is the consumer Audience field any other SAML consumer; Scheme Setup tab of the Authentication Scheme Properties dialog</p>	<p>Audience field Assertions tab of the Affiliate Properties dialog</p>
<p>Assertion Consumer URL field (SAML POST auth. scheme only) Scheme Setup tab of the Authentication Scheme Properties dialog</p>	<p>Assertion Consumer URL field Assertions tab of the Affiliate Properties dialog SMCONSUMERURL query parameter intersite transfer URL links at the producer</p>
<p>Issuer field Scheme Setup tab--Authentication Scheme Properties dialog</p>	<p>AssertionIssuerID parameter AMAssertionGenerator.properties file at the producer</p>
<p>Version from SAML Version drop-down list (SAML Artifact auth. scheme only)</p>	<p>Version from SAML Version drop-down list Assertions tab--Affiliate Properties dialog</p>
<p>Company Source ID field (SAML Artifact auth. scheme only)</p>	<p>SourceID parameter AMAssertionGenerator.properties file at the producer</p>

SAML 2.0 Matching Configuration Settings

The following table lists SiteMinder configuration settings that must be set to the same value at the SAML 2.0 Identity Provider and Service Provider. The table also indicates the dialog or file where these settings are located. Most of these settings are in the FSS Administrative UI; however, some parameters are in a properties file or part of a link.

Important! If you have to enter a URL as a value for a setting, the URL string that comes after the colon, for example, "http:" is case sensitive. Therefore, the case of all SP ID- and IdP ID-related settings must match.

These Settings at the Service Provider...	Must Match These Settings at the Identity Provider...
Attribute Name Add/Edit Attribute dialog accessed from the Attributes tab of the SAML 2.0 Auth. Scheme Properties dialog	Variable Name Attribute Fields section--SAML Service Provider Attribute dialog
Audience field any other SAML Service Provider; SSO tab of the SAML 2.0 Auth Scheme Properties dialog	Audience field SSO Tab--SAML Service Provider dialog
IdP ID field Scheme Setup tab--Authentication Scheme Properties dialog	IdP ID field General tab--Service Provider dialog For Identity Provider-initiated SSO-- SPID query parameter in an unsolicited response
Local Name field Add/Edit Attribute dialog accessed from the Attributes tab of the SAML 2.0 Auth. Scheme Properties dialog Local Name Federation Attribute Variable Properties dialog for creating a Federation Attribute variable at the SAML Requester (Service Provider).	None
SP ID field Scheme Setup tab--Authentication Scheme Properties dialog For Service Provider-initiated SSO-- ProviderID query parameter in hard-coded links to the Identity Provider	SP ID field General tab--Service Provider dialog

These Settings at the Service Provider...	Must Match These Settings at the Identity Provider...
SP Name field Backchannel tab of the SAML 2.0 Auth Scheme Properties dialog This value must be in lowercase.	Name field Service Provider dialog This value must be in lowercase.

WS-Federation Configuration Settings

The following table lists SiteMinder configuration settings that must be set to the same value at the WS-Federation Account Partner and Resource Partner. Read the table as follows:

- The first column, "Setting at Resource Partner", describes a setting that must be configured in the FSS Administrative UI at the Resource Partner.
- The second column, "Setting at the Account Partner", describes a setting that must be configured in the FSS Administrative UI at the Account Partner and must match the setting at the consumer.
- The third column, "Other Settings Requiring Matching Value", describes other configuration settings (at the Resource or Account Partner, as specified) that also require a matching setting

Important! If you have to enter a URL as a value for a setting, the URL string that comes after the colon, for example, "http:" is case sensitive. Therefore, the case of all RP ID- and AP ID-related settings must match.

These Settings at the Resource Partner...	Must Match These Settings at the Account Partner...
Resource Partner ID Scheme Setup tab--Authentication Scheme Properties dialog	Resource Partner ID field General tab Resource Partner Properties dialog wtrealm query parameter must be set to Resource Partner ID for the hard-coded link to trigger Account Partner-initiated SSO.
Account Partner ID Scheme Setup tab--Authentication Scheme Properties dialog	Account Partner ID field General tab of the Resource Partner Properties dialog

Chapter 23: Federation Web Services URLs Used by SiteMinder

This section contains the following topics:

[Federation Services URLs](#) (see page 477)

[URLs for Services at the Asserting Party](#) (see page 477)

[URLs for Services at the Relying Party](#) (see page 487)

[The Web.xml File](#) (see page 494)

Federation Services URLs

The Federation Web Services application installed by the Web Agent Option Pack or the SPS federation gateway contains many services to implement SiteMinder Federation Security Services. When configuring single sign-on, single logout, or identity provider discovery profile through the FSS Administrative UI, you are required to specify URLs that reference the different services.

The following service descriptions each include:

- A brief description of the service
- The URL for the service
- The field in the FSS Administrative UI where you enter the URL
- Associated servlet and servlet mapping in the Web.xml file

The Web.xml file is one of the deployment descriptors for the Federation Web Services application. This file lists servlets and URL mappings.

URLs for Services at the Asserting Party

The Federation Web Services application supplies the following services:

- [Intersite Transfer Service](#) (see page 478) (SAML 1.x producer)
- [Assertion Retrieval Service](#) (see page 479) (SAML 1.x producer)
- [Artifact Resolution Service](#) (see page 480) (SAML 2.0 IdP)
- [Single sign-on Service](#) (see page 481) (SAML 2.0 IdP)
- [Single logout Service](#) (see page 483) (SAML 2.0 IdP)

- [Identity Provider Discovery Profile Service](#) (see page 485) (SAML 2.0)
- [Attribute Service](#) (see page 485) (SAML 2.0)
- [Single sign-on Service](#) (see page 484) (WS-Federation)
- [Signout Service](#) (see page 482) (WS-Federation)
- [WSFedDispatcher Service](#) (see page 487) (WS-Federation)

Intersite Transfer Service URL (SAML 1.x)

For SAML 1.x POST and artifact profiles, the intersite transfer URL is a producer-side component that transfers a user from the producer to the consumer.

Default URL for this Service

`http://producer_server:port/affwebservices/public/intersitetransfer`

producer_server:port

Identifies the web server and port number of the system at the producer hosting the Web Agent Option Pack or the SPS federation gateway.

Intersite Transfer URL

Include the URL in a hard-coded link on a page at the producer.

Associated Servlet and Servlet Mapping in the Web.xml file

```
<servlet>
  <servlet-name>intersiteTransferService</servlet-name>
  <display-name>Intersite Transfer Service</display-name>
  <description>This servlet acts as the Intersite Transfer URL.</description>
  <servlet-class>com.netegrity.affiliateminder.webservices.
    IntersiteTransferService
  </servlet-class>
</servlet>

<servlet-mapping>
  <servlet-name>intersiteTransferService</servlet-name>
  <url-pattern>/public/intersitetransfer/*</url-pattern>
</servlet-mapping>
```

Assertion Retrieval Service (SAML 1.x)

The Assertion Retrieval Service retrieves an assertion for a SAML 1.x consumer site.

- Default URLs for this Service:
 - If you are using Basic or Basic over SSL to protect this service, the URL is:
https://producer_server:port/affwebservices/assertionretriever
 - If you are using client certificate authentication to protect this service, the URL is:
https://producer_server:port/affwebservices/certassertionretriever

producer_server:port

Identifies the web server and port number of the system at the producer hosting the Web Agent Option Pack or the SPS federation gateway.

- Field Where URL Entered: Assertion Retrieval URL

The Assertion Retrieval URL field is on the SAML Artifact Template dialog.

- Associated Servlet and Servlet Mapping in Web.xml file

```
<servlet>
  <servlet-name>assertionretriever</servlet-name>
  <display-name>SAML Assertion Retrieval servlet</display-name>
  <description>This servlet processes the HTTP post based SAML requests and
  returns the SAML Response elements. Both SAML Request and Response elements are
  SOAP encoded.</description>
  <servlet-class>com.netegrity.affiliateminder.webservices.
    AssertionRetriever</servlet-class>
</servlet>

<servlet-mapping>
  <servlet-name>assertionretriever</servlet-name>
  <url-pattern>/assertionretriever/*</url-pattern>
</servlet-mapping>
<servlet-mapping>

  <servlet-name>assertionretriever</servlet-name>
  <url-pattern>/certassertionretriever/*</url-pattern>
</servlet-mapping>
```

Artifact Resolution Service (SAML 2.0)

The Artifact Resolution Service is used to retrieve SAML 2.0 assertions for a Service Provider.

- Default URL for this Service:
 - If you are using Basic authentication to protect this service, the URL is:
`http://idp_server:port/affwebservices/saml2artifactresolution`
 - If you are using Basic over SSL or X.509 client certificate authentication to protect this service, the URL is:
`https://idp_server:port/affwebservices/saml2certartifactresolution`

idp_server:port

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

- Field Where URL Entered: Resolution Service

The Resolution Service field is on the SSO tab of the SAML 2.0 Auth. Scheme Properties dialog. Select HTTP-Artifact to make the field active.

- Associated Servlet and Servlet Mapping in Web.xml file

```
<servlet>
  <servlet-name>saml2artifactresolution</servlet-name>
  <display-name>SAML 2.0 Single Sign-On service</display-name>
  <description>This servlet is the SAML 2.0 Artifact Resolution
    service at an IdP.</description>
  <servlet-class>com.netegrity.affiliateminder.webservices.
    saml2.ArtifactResolution</servlet-class>
</servlet>
```

```
<servlet-mapping>
<servlet-name>saml2artifactresolution</servlet-name>
<url-pattern>/saml2artifactresolution/*</url-pattern>
</servlet-mapping>
```

```
<servlet-mapping>
<servlet-name>saml2artifactresolution</servlet-name>
<url-pattern>/saml2certartifactresolution/*</url-pattern>
</servlet-mapping>
```


Single Sign On Service (SAML 2.0)

Implements single sign-on for SAML 2.0.

- Default URL for this Service:

`http://idp_server:port/affwebservices/public/saml2sso`

idp_server:port

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

- Field Where URL Entered: SSO Service

This SSO Service field is in the SSO tab of the SAML 2.0 Auth. Scheme Properties dialog.

- Associated Servlet and Servlet Mapping in Web.xml file

```
<servlet>
  <servlet-name>saml2sso</servlet-name>
  <display-name>SAML 2.0 Single Sign-On service</display-name>
  <description>This servlet is the SAML 2.0 Single Sign-On service at an
  IdP.</description>
  <servlet-class>com.netegrity.affiliateminder.webservices.
  saml2.SSO</servlet-class>
</servlet>

<servlet-mapping>
  <servlet-name>saml2sso</servlet-name>
  <url-pattern>/public/saml2sso/*</url-pattern>
</servlet-mapping>
```

Single Sign-on Service (WS-Federation)

Implements single sign-on for WS-Federation.

- Default URL for this Service:

`http://ap_server:port/affwebservices/public/wsfedsso`

ap_server:port

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

- Field Where URL Entered: SSO Service

This SSO Service field is in the SSO tab of the WS-Federation Auth. Scheme Properties dialog.

- Associated Servlet and Servlet Mapping in Web.xml file

```
<servlet>
<servlet-name>wsfedsso</servlet-name>
<display-name>WSFED Single Sign-On service</display-name>
<description>This servlet is the WSFED Single Sign-On service at an Account
Partner.</description>
<servlet-class>com.netegrity.affiliateminder.webservices.wsfed.SSO
  </servlet-class>
</servlet>
```

```
<servlet-mapping>
<servlet-name>wsfedsso</servlet-name>
<url-pattern>/public/wsfedsso/*</url-pattern>
</servlet-mapping>
```

Single Logout Service at the IdP (SAML 2.0)

This service implements single logout for SAML 2.0.

- Default URL for this Service:

`http://idp_server:port/affwebservices/public/saml2slo`

idp_server:port

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

- Fields Where URL Entered: SLO Location URL and the SLO Response Location URL

At the Identity Provider, these fields are on the SLO tab of the Service Provider Properties dialog.

- Associated Servlet and Servlet Mapping in Web.xml file

```
<servlet>
  <servlet-name>saml2slo</servlet-name>
  <display-name>SAML 2.0 Single Logout service</display-name>
  <description>This servlet is the SAML 2.0 Single Logout service at an
  IdP.</description>
  <servlet-class>com.netegrity.affiliateminder.webservices.
    saml2.SLOService</servlet-class>
</servlet>

<servlet-mapping>
  <servlet-name>saml2slo</servlet-name>
  <url-pattern>/public/saml2slo/*</url-pattern>
</servlet-mapping>
```

Signout Service at the AP (WS-Federation)

This service implements sign out service for WS-Federation.

- Default URL for this Service:

`http://ap_server:port/affwebservices/public/wsfedsignout`

ap_server:port

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

- Fields Where URL Entered: Signout Cleanup URL and the Signout URL

At the Account Partner, these fields are on the Signout tab of the Resource Partner Properties dialog.

- Associated Servlet and Servlet Mapping in Web.xml file

```
<ervlet>
  <ervlet-name>wsfedsignout</ervlet-name>
  <display-name>WS-Federation Signout Service</display-name>
  <description>This servlet is the WS-Federation Signout service
    at an AP.</description>
  <ervlet-class>com.netegrity.affiliateminder.webservices.wsfed.
    SignoutService</ervlet-class>
</ervlet>
```

```
<ervlet-mapping>
<ervlet-name>ws fedsignout</ervlet-name>
<url-pattern>/public/wsfedsignout/*</url-pattern>
</ervlet-mapping>
```

Identity Provider Discovery Profile Service (SAML 2.0)

This service implements the Identity Provider Discovery Profile.

- Default URL for this Service:

`https://idp_server:port/affwebservices/public/saml2ipd/*`

idp_server:port

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

- Field Where URL Entered: Service URL

This Service URL is on the IPD tab in the SAML Service Provider Properties dialog.

- Associated Servlet and Servlet Mapping in Web.xml file

```
<servlet>
  <servlet-name>saml2ipd</servlet-name>
  <display-name>SAML 2.X Identity Provider Discovery Profile
    service</display-name>
  <description>This servlet is the SAML 2.X Identity Provider Discovery Profile
    service at an SP or IdP.</description>
  <servlet-class>com.netegrity.affiliateminder.webservices.
    saml2.IPDServlet</servlet-class>
</servlet>

<servlet-mapping>
  <servlet-name>saml2ipd</servlet-name>
  <url-pattern>/public/saml2ipd/*</url-pattern>
</servlet-mapping>
```

Attribute Service (SAML 2.0)

The Attribute Service enables an Identity Provider acting as an Attribute Authority to respond to attribute queries from a Service Provider acting as a SAML Requester.

- Default URL for this Service:

`http://idp_server:port/affwebservices/saml2attributeservice`

idp_server:port

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

- Fields Where URL Entered:

At the Service Provider, the Attributes tab of the SAML 2.0 Auth. Scheme Properties dialog.

- Associated Servlet and Servlet Mapping in the Web.xml file

```
<servlet>
  <servlet-name>saml2attributeservice</servlet-name>
  <display-name>SAML 2.0 Attribute service</display-name>
  <description>This servlet is the SAML 2.0 Attribute Service
    at an IdP.</description>
  <servlet-class>com.netegrity.affiliateminder.webservices.saml2.
    AttributeService</servlet-class>
</servlet>

<servlet-mapping>
  <servlet-name>saml2attributeservice</servlet-name>
  <url-pattern>/saml2attributeservice/*</url-pattern>
</servlet-mapping>

<servlet-mapping>
  <servlet-name>saml2attributeservice</servlet-name>
  <url-pattern>/saml2certattributeservice/*</url-pattern>
</servlet-mapping>
```

WSFedDispatcher Service at the AP

The WSFedDispatcher Service receives all incoming WS-Federation messages and forwards the request processing to other services based on the query parameter data.

- Default URL for this Service:

`https://ap_server:port/affwebservices/public/wsfeddispatcher`

ap_server:port

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

- Field Where URL Entered: Not applicable
- Associated Servlet and Servlet Mapping in Web.xml file

```
<servlet>
  <servlet-name>wsfeddispatcher</servlet-name>
  <display-name>WS-Federation Dispatcher service</display-name>
  <description>This servlet is the WS-Federation Dispatcher service for all
WS-Federation services.</description>
  <servlet-class>com.netegrity.affiliateminder.webservices.wsfed.
    dispatcher</servlet-class>
</servlet>

<<servlet-mapping>
<servlet-name>wsfeddispatcher</servlet-name>
<url-pattern>/public/wsfeddispatcher/*</url-pattern>
</servlet-mapping>
```

URLs for Services at the Relying Party

The relying party provides the following services; however, you enter the URL for the service at the asserting party.

The SiteMinder relying party provides the following services:

- SAML credential collector (SAML 1.x)
- AuthnRequest service (SAML 2.0)
- Assertion Consumer Service (SAML 2.0)
- Security Token Consumer Service (WS-Federation)

- Single Logout Service (SAML 2.0)
- Signout Service (WS-Federation)
- WSFedDispatcher Service (WS-Federation)

SAML Credential Collector (SAML 1.x)

This service assists in consuming the SAML 1.x assertion.

- Default URL for this Service:

`https://consumer_server:port/affwebservices/public/samlcc`

consumer_server:port

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

- Field Where URL Entered: Assertion Consumer URL

This field is on the Assertions tab of the Affiliate Properties dialog and the SAML POST Template dialog.

- Associated Servlet and Servlet Mapping in Web.xml file

```
<servlet>
  <servlet-name>samlcredentialcollector</servlet-name>
  <display-name>SAML Credential Collector</display-name>
  <description>This servlet acts as the SAML Credential Collector.</description>
  <servlet-class>com.netegrity.affiliateminder.webservices.
    SAMLCredentialCollector</servlet-class>
</servlet>

<servlet-mapping>
  <servlet-name>samlcredentialcollector</servlet-name>
  <url-pattern>/public/samlcc/*</url-pattern>
</servlet-mapping>
```


AuthnRequest (SAML 2.0)

This service helps implement single sign-on for artifact or POST profile.

- Default URL for this Service:

`https://sp_server:port/affwebservices/public/saml2authnrequest`

sp_server:port

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

- Field Where URL Entered: Not applicable.

The authnrequest is a link in an application at the Service Provider. This link initiates single sign-on and must be included in an application.

- Associated Servlet and Servlet Mapping in Web.xml file

```
<servlet>
  <servlet-name>saml2authnrequest</servlet-name>
  <display-name>SAML 2.0 AuthnRequest service</display-name>
  <description>This servlet is the SAML 2.0 AuthnRequest service at an
SP.</description>
  <servlet-class>com.netegrity.affiliateminder.webservices.
    saml2.AuthnRequest</servlet-class>
</servlet>

<servlet-mapping>
  <servlet-name>saml2authnrequest</servlet-name>
  <url-pattern>/public/saml2authnrequest/*</url-pattern>
</servlet-mapping>
```

Assertion Consumer Service (SAML 2.0)

This service enables the consumption of assertions.

- Default URL for this Service:

`https://sp_server:port/affwebservices/public/saml2assertionconsumer`

sp_server:port

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

- Field Where URL Entered: Assertion Consumer URL

This Assertion Consumer URL is on the SSO tab of the SAML Service Provider Properties dialog at the Identity Provider.

- Associated Servlet and Servlet Mapping in Web.xml file

```
<servlet>
  <servlet-name>saml2assertionconsumer</servlet-name>
  <display-name>SAML 2.0 Assertion Consumer service</display-name>
  <description>This servlet is the SAML 2.0 Assertion Consumer service at an
  SP.</description>
  <servlet-class>com.netegrity.affiliateminder.webservices.
    saml2.AssertionConsumer</servlet-class>
</servlet>

<servlet-mapping>
  <servlet-name>saml2assertionconsumer</servlet-name>
  <url-pattern>/public/saml2assertionconsumer/*</url-pattern>
</servlet-mapping>
```

Security Token Consumer Service (WS-Federation)

The Security Token Consumer Service enables the consumption of assertions at the Resource Partner.

- Default URL for this Service:

`https://rp_server:port/affwebservices/public/wsfedsecuritytokenconsumer`

rp_server:port

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

- Field Where URL Entered: Security Token Consumer Service

This Security Token Consumer Service field is on the SSO tab of the Resource Partner Properties dialog.

- Associated Servlet and Servlet Mapping in Web.xml file

```
<servlet>
  <servlet-name>wsfedsecuritytokenconsumer</servlet-name>
  <display-name>Security Token Consumer service</display-name>
  <description>This servlet is the WS-Federation Security Token
    Consumer service at an RP.</description>
  <servlet-class>com.netegrity.affiliateminder.webservices.wsfed.
    SecurityTokenConsumer</servlet-class>
</servlet>

<<servlet-mapping>
<servlet-name>wsfedsecuritytokenconsumer</servlet-name>
<url-pattern>/public/wsfedsecuritytokenconsumer/*</url-pattern>
</servlet-mapping>
```

Single Logout Service at the SP (SAML 2.0)

This service implements single logout for SAML 2.0.

- Default URL for this Service:

`http://sp_server:port/affwebservices/public/saml2slo`

sp_server:port

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

- Fields Where URL Entered: SLO Location URL and SLO Response Location URL

At the Service Provider, these fields are on the SLO tab of the SAML 2.0 Auth. Scheme Properties dialog.

- Associated Servlet and Servlet Mapping in Web.xml file

```
<servlet>
  <servlet-name>saml2slo</servlet-name>
  <display-name>SAML 2.0 Single Logout service</display-name>
  <description>This servlet is the SAML 2.0 Single Logout service at an
  SP.</description>
  <servlet-class>com.netegrity.affiliateminder.webservices.
    saml2.SLOService</servlet-class>
</servlet>

<servlet-mapping>
  <servlet-name>saml2slo</servlet-name>
  <url-pattern>/public/saml2slo/*</url-pattern>
</servlet-mapping>
```

Signout Service at the RP (WS-Federation)

This service implements sign out service for WS-Federation.

- Default URL for this Service:

`http://rp_server:port/affwebservices/public/wsfedsignout`

rp_server:port

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

- Fields Where URL Entered: Signout Cleanup URL and Signout URL

At the Resource Partner, these fields are on the Signout tab of the WS-Federation Auth. Scheme Properties dialog.

- Associated Servlet and Servlet Mapping in Web.xml file

```
<servlet>
  <servlet-name>wsfedsignout</servlet-name>
  <display-name>WS-Federation Signout Service</display-name>
  <description>This servlet is the WS-Federation Signout service
    at an RP.</description>
  <servlet-class>com.netegrity.affiliateminder.webservices.wsfed.
    SignoutService</servlet-class>
</servlet>
```

```
<servlet-mapping>
  <servlet-name>ws fedsignout</servlet-name>
  <url-pattern>/public/wsfedsignout/*</url-pattern>
</servlet-mapping>
```

WSFedDispatcher Service at the RP

The WSFedDispatcher Service services receives all incoming WS-Federation messages and forwards the request processing to other services based on the query parameter data.

- Default URL for this Service:

`https://rp_server:port/affwebservices/public/wsfeddispatcher`

rp_server:port

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

- Field Where URL Entered: Not applicable
- Associated Servlet and Servlet Mapping in Web.xml file

```
<servlet>
  <servlet-name>wsfeddispatcher</servlet-name>
  <display-name>WS-Federation Dispatcher service</display-name>
  <description>This servlet is the WS-Federation Dispatcher service for all
WS-Federation services.</description>
  <servlet-class>com.netegrity.affiliateminder.webservices.wsfed.
  dispatcher</servlet-class>
</servlet>

<<servlet-mapping>
<servlet-name>wsfeddispatcher</servlet-name>
<url-pattern>/public/wsfeddispatcher/*</url-pattern>
</servlet-mapping>
```

The Web.xml File

The Web.xml file lists servlets and URL mappings for the Federation Web Services application.

You cannot change most of this file, but you can modify the URL mappings.

To view the Web.xml file, go to the appropriate file location:

- `web_agent_home/affwebservices/WEB-INF`
- `sps_home/secure-proxy/Tomcat/webapps/affwebservices/WEB-INF`

Chapter 24: Troubleshooting

This section contains the following topics:

[General Issues](#) (see page 495)

[SAML 1.x-Only Issues](#) (see page 501)

[SAML 2.0-Only Issues](#) (see page 503)

General Issues

The following troubleshooting topics apply to SAML 1.x and SAML 2.0.

Web Agent Option Pack Fails to Initialize Due to Invalid smjavaagent.dll

Symptom:

The Web Agent Option Pack fails to initialize with on a system with other CA products. Error messages, such as "Java Agent API initialization FAILED" or "unsatisfied link error" display.

Error messages similar to the following appear in the Federation Web Service log file:

```
11:04:46 AM[29959477:E] Exception while reading the WebAgent configuration
information: javaagent_api_getConfig
11:04:46 AM[29959477:E] Java Agent API initialization FAILED.
```

Solution:

An invalid version of smjavaagentapi.dll can be present the system path. Verify that all installed products are compatible with one another and of compatible versions.

To verify the versions

1. Log in to the [Technical Support site](#).
2. Search for the SiteMinder Platform Support Matrix for r12.0 SP3.

Cookie Domain Mismatch Errors

Symptom:

After successful SAML authentication at consumer/SP site, the consumer/SP Web Agent still challenges the user because of cookie domain mismatch.

Solution:

Verify that the producer/IdP and consumer/SP are not in the same cookie domain. Legacy federation does not support federation within the same cookie domain. Separate cookie domains are required at the producer/IdP and consumer/SP sites. Additionally, verify that the CookieDomainScope parameter is set to the appropriate value for your environment. This parameter is a Web Agent parameter (see information about single sign-on in the *SiteMinder Web Agent Configuration Guide*).

If separate cookie domains are in use, verify that the cookie domain in the Agent configuration matches the domain name in the requested target URL.

Error After Successful Authentication at Consumer/SP

Symptom:

After successful authentication at the consumer site, an HTTP 404 "Page Not Found" error code is returned to the browser.

Solution:

Verify that the target page exists in the web server document root. Examine the FWS trace log to verify that the user is being redirected to the correct URL.

HTTP 404 Error When Trying to Retrieve Assertion at the Consumer

Symptom:

When the relying party tries to retrieve an assertion, an HTTP 404 "Page Not Found" error code is returned to the browser.

Solution:

Verify that the Federation Web Services application is deployed as a web application. Deploy the application on a web server running one of the supported application servers. The SiteMinder Platform Support Matrix lists the supported platforms for the Web Agent Option Pack.

Federation Web Services Fails to Send SAML Request to Producer/IdP

Symptom:

The Federation Web Services application at the consumer/SP fails to send a SAML request message to the producer/IdP. The consuming side fails to trust the certificate of the web server.

Solution:

Add the certificate of the Certificate Authority that issued the client certificate to the key database of the web server at the producer/IdP.

Matching Parameter Case-Sensitivity Configuration Issues

Symptom:

Problems occur due to conflicts between configuration parameters that must correspond on producer/Identity Provider and consumer/Service Provider, even though the parameters appear to match.

Solution:

The URL string that comes after the colon is case-sensitive. For example, the text after **http:** is case-sensitive. Therefore, the case of the URLs in all corresponding settings must match.

Parameter values that must match between the asserting and relying parties are documented in the topic [Configuration Settings that Must Use the Same Values](#) (see page 471).

Error Message When Viewing FederationWSCustomUserStore

Symptom:

On Windows 2000 SP4 Japanese OS system, SiteMinder displays the error, "Search operation failed: SiteMinder Administration Error: User directory error. (Error 60)" when you display the properties of the FederationWSCustomUserStore user directory and click the View Contents button. This error occurs if no affiliate domain has been created.

Solution:

Verify that an affiliate domain has been created.

Policy Server System Fails After Logoff

Symptom:

In some environments, logging off the Policy Server while it is running causes the Policy Server to fail. The failure is due to a JVM issue.

Solution:

Add the `-Xrs` command to its own command line in the `JVMOptions.txt` file. This command is case-sensitive, so add it as shown. This command reduces usage of operating system signals by the JVM.

The `JVMOptions.txt` file is located in `policy_server_home/config/`.

Encrypted Private Key Fails to Be Imported into SMkeydatabase

Symptom:

When you try to import an encrypted private key in the `smkeydatabase`, the import fails.

Solution:

Private keys can be encrypted using only the following encryption algorithms:

- PKCS#12: PBE-SHA1-RC4-128, PBE-SHA1-RC4-40, PBE-SHA1-3DES, PBE-SHA1-2DES, PBE-SHA1-RC2-128, PBE-SHA1-RC2-40
- PKCS#5 v1 PBE-MD5-DES

Multibyte Characters in Assertions are Not Handled Properly

Symptom:

When you include a multibyte character in an assertion, problems can occur.

Solution:

Set the `LANG` setting for your operating system to UTF-8, as follows:

```
LANG=xx_xx.UTF-8
```

For example, for Japanese, the entry would be:

```
LANG=ja_JP.UTF-8
```

Trace Logs Not Appearing for IIS Web Server Using ServletExec

Symptom:

You have enabled trace logging in the LoggerConfig.properties file, but the affwebservices.log and FWStrace.log files are not being written to the WEB-INF/classes directory.

Solution:

Verifies that the anonymous user account associated with ServletExec has permissions to write to the Windows file system. If the user account does not have the right to act as part of the operating system, ServletExec cannot write the log files.

Error During Initialization of JVM

Symptom:

If you receive the following error message in the Policy Server log (figure out which log):
Error occurred during initialization of JVM
Could not reserve enough space for object heap.

The Web Agent Option Pack functionality is not working due to a JVM initialization failure.

Solution:

Restrict the object heap memory size.

To restrict the memory size

1. Open the JVMOptions.txt file, in the directory *web_agent_home*/WEB-INF/properties file.
2. Add the following entry to the file as it is written here:
`-Xms128M`
3. Save the file.
4. Restart the Policy Server.

Affwebserver.log and FWSTrace.log Show Wrong Time

Symptom:

If the Web Agent Option Pack is installed on a Weblogic 8.1.6 server, the time stamps in the affwebservices.log and the FWSTrace.log are in GMT time not local time, despite the LogLocalTime parameter in the LoggerConfig.properties file being set to Yes.

Solution:

WebLogic is referencing the LoggerConfig properties file in a different location than the default location of the SAML Affiliate Agent.

To verify that the log files use local time

1. Navigate to the LoggerConfig.properties file at the following location:
`\bea\user_projects\domains\ca_apps_domain\FWS\stage\affwebservices\affwebservices\WEB-INF\classes\LoggerConfig.properties`
2. Open the properties file and set the LogLocalTime parameter to Yes.
3. Restart the WebLogic server.

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.

SAML 1.x-Only Issues

The following issues apply only to SAML 1.x features.

SAML 1.x Artifact Profile Single Sign-On Failing

Symptom:

If single sign-on with the SAML 1.x artifact profile is configured, the consumer site fails to send SAML request messages to the producer. Error messages similar to the following appear in the Federation Web Service log file:

```
May 23, 2012 4:20:44.234 PM[28349544:E] Dispatcher object thrown unknown exception while processing the request message. Message: java.net.ConnectException: Connection refused: connect.
```

```
May 23, 2012 4:20:44.234 PM[28349544:E] Exception caught. Message: com.netegrity.affiliateminder.webservices.m: Exception occurred while message dispatcher(srca) object trying to send SOAP request message to the SAML producer.
```

Solution:

Verify that the web server hosting the Assertion Retrieval Service is running with a configured SSL port.

Consumer Not Authenticating When Accessing Assertion Retrieval Service

Symptom:

In an environment using SAML 1.x artifact single sign-on, the consumer fails authentication when trying to access the Assertion Retrieval Service at the producer.

Solution:

Depends upon the configured authentication:

- If Basic authentication is configured to protect the Assertion Retrieval Service, verify that the Name and Password values in the Affiliate configuration match the Affiliate Name and Password values configured for the SAML Artifact authentication scheme.
- If client certificate authentication is configured to protect the Assertion Retrieval Service, verify that the client certificate of the consumer is valid and that it is present in the AM.keystore database of the consumer. Additionally, verify that the certificate of the Certificate Authority that issued the client certificate is present in the web server key database at the producer.

Authentication Fails After Modifying Authentication Method

Symptom:

If you change the authentication method protecting the SAML 1.x Assertion Retrieval Service from Basic to Client Cert, subsequent authentication requests can fail.

If you change the authentication method protecting the SAML 1.x Assertion Retrieval Service from Client Cert to Basic, subsequent authentication requests can fail.

Solution:

Restart the web server after the authentication method is changed.

Client Authentication Fails for SAML Artifact Single Sign-on

Symptom:

Client certificate authentication for SAML 1.x artifact single sign-on fails at the producer. The following error is logged in the web agent trace logs:

```
Setting HTTP response variable HTTP_consumer_name=from SiteMinder
```

For example, if the Attribute Name in the response is configured as "name" for an LDAP User Directory, the response fails.

Solution:

Verify that you create a Web Agent response under the domain FederationWebServicesDomain. The response must be as follows:

Attribute type

WebAgent HTTP Header variable

Attribute Kind

User Attribute

Variable Name

consumer_name

Attribute Name

uid (for LDAP) or name (for ODBC)

SAML 2.0-Only Issues

The following issues apply only to SAML 2.0 features.

SP Not Authenticating When Accessing Assertion Retrieval Service

Symptom:

In an environment using SAML 2.0 artifact single sign-on, the Service Provider fails to authenticate when attempting to access the Artifact Resolution Service at the Identity Provider.

Error messages similar to the following appear in the Federation Web Service log file:
May 23, 2005 4:43:51.479 PM[31538514:E] SAML producer returned error http status code.
HTTP return status: 401. Message: <HTML><HEAD><TITLE>401: Access Denied</TITLE></HEAD><BODY><H1>401: Access Denied</H1>
Proper authorization is required for this area. Either your browser does not perform authorization, or your authorization has failed.</BODY></HTML>

Solution:

Depends upon the configured authentication:

- If Basic authentication is configured, verify that the Name and Password values specified in the Service Provider Properties dialog at the IdP match the Affiliate Name and Password values configured for the SAML 2.0 authentication scheme at the SP.
- If client certificate authentication is configured to protect the Artifact Resolution Service, verify that the client certificate of the Service Provider is valid and that it is in the AM.keystore database of the Service Provider. Additionally, verify that the Certificate Authority that issued the client certificate is in the own key database of the web server at the Identity Provider.
- If no authentication is configured, verify that the Artifact Resolution Service URL is *not* protected.

ODBC Errors Deleting Expiry Data From Session Store

Symptom:

If you upgrade a Policy Server from an earlier version, ODBC errors can occur when deleting expiry data from the session store.

Solution:

Upgrade the session store schema as described in the *SiteMinder Upgrade Guide*.

Appendix A: Federation Security Services Process Flow

This section shows the detailed flow between the components that comprise the SiteMinder Federated Security Services. This section assumes that the reader knows SiteMinder interactions between a Web Agent and Policy Server.

This section contains the following topics:

[Flow Diagram for SSO Using SAML 1.x Artifact Authentication](#) (see page 505)

[Flow Diagram for SSO Using SAML 1.x POST Profile Authentication](#) (see page 508)

[Flow Diagram for SSO Using SAML 2.0 Authentication with Artifact Binding](#) (see page 510)

[Flow Diagram for SSO Using SAML 2.0 Authentication with POST Binding](#) (see page 514)

[Flow Diagram for WS-Federation SSO Initiated at the Resource Partner](#) (see page 518)

[Flow Diagram for SAML 2.0 Single Logout](#) (see page 522)

[Flow Diagram for WS-Federation Signout \(AP-initiated\)](#) (see page 525)

[Flow Diagram for WS-Federation Signout \(RP-initiated\)](#) (see page 528)

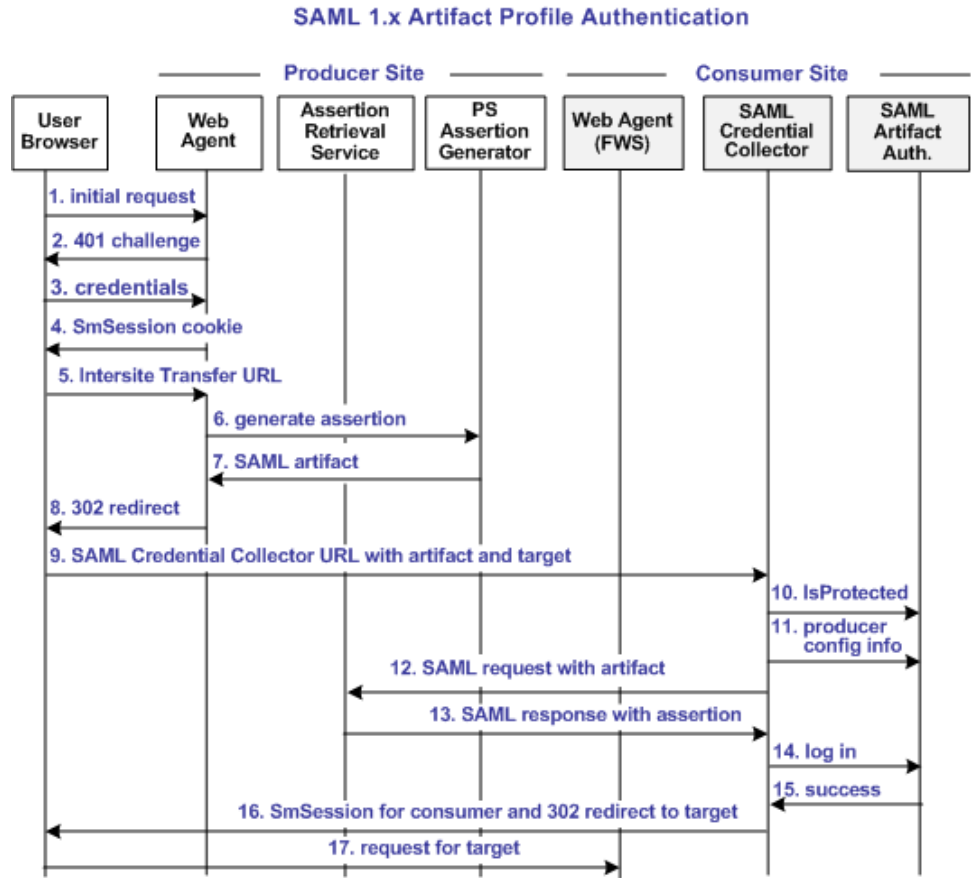
[Flow Diagram for Identity Provider Discovery Profile](#) (see page 530)

Flow Diagram for SSO Using SAML 1.x Artifact Authentication

The following illustration shows the flow between a user and the Federation Security Services components at the producer and consumer sites. This set-up enables single sign-on between the sites. SAML artifact profile is the authentication method and the flow diagram assumes successful authentication and authorization at the producer and consumer sites.

Note: This flow applies to examples that do not use the SAML Affiliate Agent.

The process flow diagram for SAML 1.x Artifact Authentication follows.



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

The sequence of actions is as follows:

1. The user makes an initial request to a protected page at the producer site.
2. The Web Agent at the producer site responds with a 401 challenge to the user.
3. The user submits credentials, such as the user name and password to the Web Agent.
4. The Web Agent issues a SiteMinder SMSESSION cookie to the browser of the user for the producer site domain.

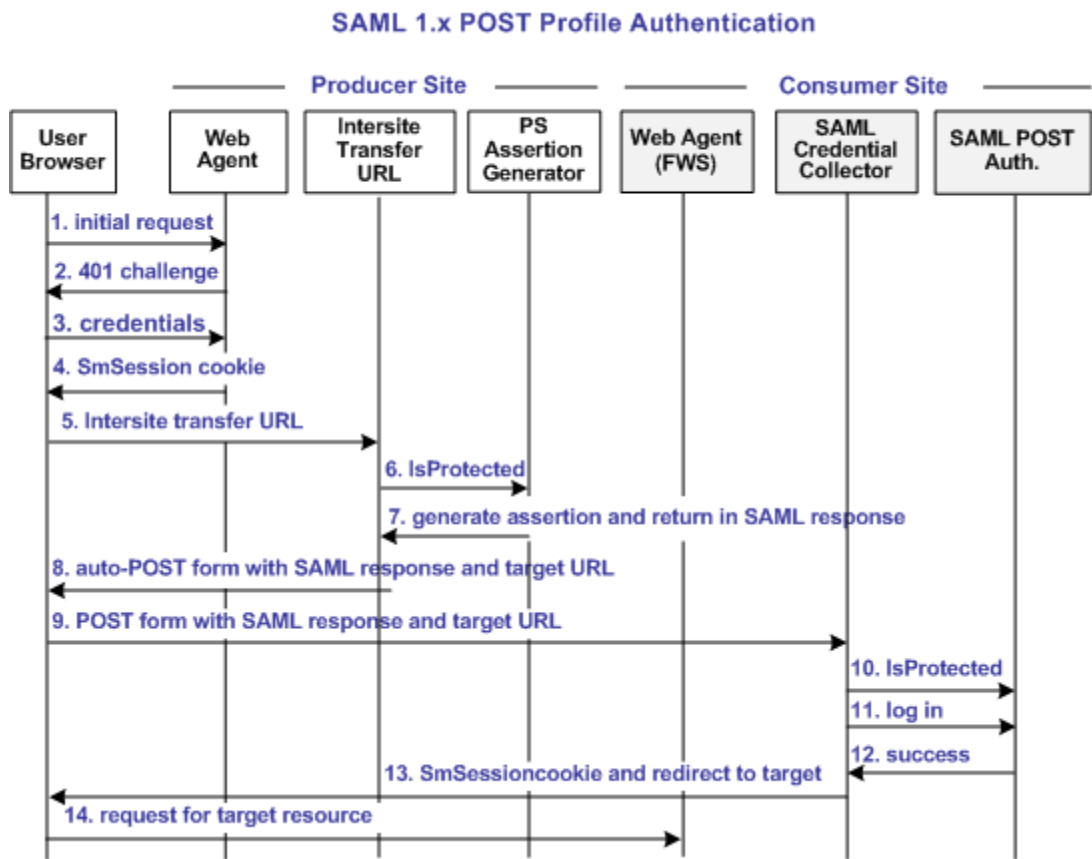
5. The user clicks a link to visit the consumer site. This link is referred to as the intersite transfer URL because it results in transferring the user to another site. The intersite transfer URL makes a request to the Web Agent at the producer site first. This URL contains the location of the SAML credential collector and the target URL to access at the consumer site.
6. The Web Agent at the producer site handles the intersite transfer URL request by calling the assertion generator.
7. The assertion generator generates a SAML assertion, places it in the session store and returns the SAML artifact for the assertion.
8. The Web Agent responds with a 302 redirect to the SAML credential collector at the consumer. The redirect contains the SAML artifact and the target URL as query parameters.
9. The browser makes a request to the URL for the SAML credential collector at the consumer site.
10. The SAML credential collector handles the URL request by making an isProtected call to the SAML artifact authentication scheme.
11. The SAML artifact authentication scheme returns the producer configuration information.
12. The SAML credential collector uses the producer configuration information to make a SAML request to the assertion retrieval service at the producer. In this step, the SAML credential collector is acting as an HTTP client.
13. The assertion retrieval service at the producer retrieves the SAML assertion from the session store. The service responds with a SAML response that contains the SAML assertion.
14. The SAML credential collector makes a login call to the SAML artifact authentication scheme, passing the SAML assertion as credentials.
15. The SAML artifact authentication scheme validates the SAML assertion. The authentication scheme looks up the user record. The lookup is based on the user mapping that is configured for the scheme. The scheme returns a success reply. If the SAML assertion is not valid or a user record cannot be located, the scheme returns a failure reply.
16. If the scheme returns a success reply, the SAML credential collector issues a SiteMinder SMSESSION cookie for the consumer domain to the browser. The SAML credential collector also issues a 302 redirect to the target URL. If the scheme returns a failure reply, the SAML credential collector issue a 302 redirect to a no access URL.
17. The browser makes a request to the target URL at the consumer, which the Web Agent protects.

Flow Diagram for SSO Using SAML 1.x POST Profile Authentication

The following illustration shows the detailed flow between a user and the components at producer and consumer sites. This set-up enables single sign-on between the sites. SAML POST profile is the authentication method and the illustration assumes successful authentication and authorization at the producer and consumer sites.

Note: This flow applies to examples that do not use the SAML Affiliate Agent.

The process flow diagram for SAML 1.x POST Profile follows.



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

The sequence of events is as follows:

1. User requests a local page at the producer, which the Web Agent protects.
2. The Web Agent at the producer asks for user credentials.

This flow diagram assumes that the resource is protected with basic authentication and that user name and password are the required credentials.

3. The user submits credentials.
4. The Agent at the producer issues an SMSESSION cookie for the producer site domain and allows access to the local page.
5. The user selects a link at the local page of the producer to visit the consumer. The link looks like it goes to the consumer site, but it goes to the intersite transfer URL. The URL contains the affiliate name, the assertion consumer URL, and the target resource as query parameters.
6. The Intersite Transfer Service makes an IsProtected call to the Policy Server for the resource. The URL contains the name query parameter that uniquely identifies the consumer.
7. The Policy Server recognizes the request as a request for a SAML assertion. The Policy Server generates the assertion and returns it in a digitally signed SAML response message. The Policy Server then returns the response to the intersite transfer URL.
8. The intersite transfer URL service generates an auto-POST form containing the encoded SAML response and the target URL as form variables. The service sends the form to the browser.
9. The browser of the user automatically posts the HTML form to the SAML Credential Collector at the consumer site. This URL is read from the SAML response that the intersite transfer URL service sends.
10. The Credential Collector makes an isProtected call to the SAML POST profile authentication scheme. The authentication scheme informs the assertion consumer what type of credentials are required.
11. The Credential collector makes a login call for the requested target resource to the SAML POST profile authentication scheme, passing the assertion as credentials.
12. If the login succeeds, the SAML Credential Collector generates an SMSESSION cookie for the consumer site domain.
13. The SMSESSION cookie is placed in the browser and redirects the user to the target resource.
14. The browser requests the target resource, which the consumer-side Web Agent protects. The browser has an SMSESSION cookie for the consumer domain so the Web Agent does not challenge the user.

Flow Diagram for SSO Using SAML 2.0 Authentication with Artifact Binding

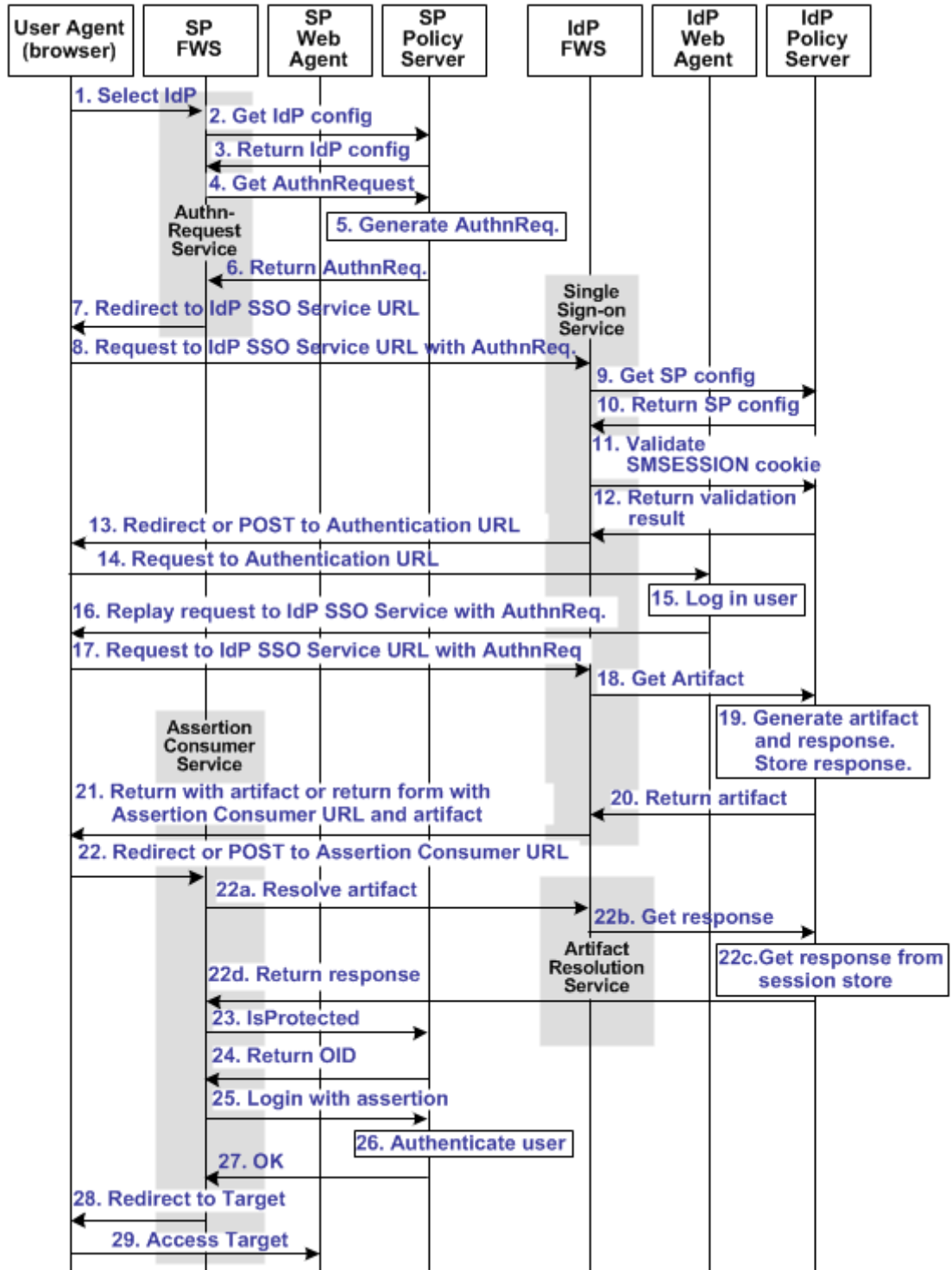
The following illustration shows the detailed flow between a user and the components at the Identity Provider and Service Provider. This set-up enables single sign-on between the sites and uses the SAML 2.0 authentication scheme with the artifact binding as the authentication method.

The flow diagram assumes the following information:

- The SP initiates the request for a resource.
- Successful authentication and authorization at the IdP and SP sites.

Note: This flow applies to examples that do not use the SAML Affiliate Agent.

The flow diagram for SAML 2.0 Authentication-Artifact Binding follows.



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

The sequence of events is as follows:

1. The user chooses a link at the SP to authenticate at a specific IdP. This link must include a Provider ID representing the chosen IdP.
2. SP FWS requests the IdP configuration information from the local Policy Server.
3. The local Policy Server returns the IdP configuration information to SP FWS. FWS can cache the configuration information.
4. SP FWS requests an AuthnRequest message from the local Policy Server through a tunnel call, passing the Provider ID. This call must contain the artifact profile in the ProtocolBinding element value.
5. The local Policy Server generates the AuthnRequest message in an HTTP redirect binding.
6. The local Policy Server returns the AuthnRequest message to the SP FWS in an HTTP redirect binding.
7. SP FWS redirects the user to the IdP SSO Service URL. FWS obtains the configuration information and includes it in the AuthnRequest message in an HTTP redirect binding.
8. The browser requests the IdP single sign-on service (SSO) URL.
9. IdP FWS requests the SP configuration information from the IdP local Policy Server.
10. The local Policy Server returns the configuration information.

Note: FWS can cache the configuration information.

11. IdP FWS gets an SMSESSION cookie for the domain of this IdP and calls the Policy Server to validate it. If there is no SMSESSION cookie, the IDP FWS skips redirects or posts to the Authentication URL.
12. The Policy Server verifies the validity of the SMSESSION cookie and returns the result.
13. If the SMSESSION cookie does not exist or is not valid, the IDP FWS redirects or posts to the Authentication URL. If the SMSESSION cookie is valid, the IDP FWS requests a SAML 2.0 artifact from the local Policy Server (see step 18).
14. The browser requests the Authentication URL, which the IdP Web Agent protects.
15. The IdP Web Agent logs the user in, setting the SMSESSION cookie, and lets the request pass to the Authentication URL.
16. The Authentication URL is the redirect.jsp file, which replays the request to the IdP single sign-on service with the AuthnRequest message.

17. The browser requests the IdP single sign-on service URL. This request is equivalent to the request from step 8, but now the user has a valid SMSESSION cookie.
18. The IdP FWS requests a SAML 2.0 artifact from the local Policy Server. FWS passes the AuthnRequest through an authorization call to the realm obtained from the configuration information.
19. The Policy Server generates the artifact and the corresponding response message. The message is formed from the configuration information from the Service Provider. The Policy Server stores the response in the session store.

The message is stored as a session variable, and is named using the string representation of the artifact message handle.
20. The Policy Server returns the artifact to IdP FWS.
21. Based on the SP configuration information, the IdP FWS takes one of the following actions:
 - Redirects the browser to the Assertion Consumer URL at the SP. The URL-encoded artifact is a URL parameter. The Assertion Consumer URL is obtained from the configuration information.
 - Returns a form containing the artifact form-encoded in two hidden form controls.

The form is wrapped into a JavaScript to auto-POST the data when the browser reads it.

Note: The assertion generator can indicate that the authentication level for the current session is too low. If the level is too low, the IdP FWS redirects to the authentication URL to facilitate step-up authentication.

22. If the artifact was sent as part of a URL, the browser redirects the user to the Assertion Consumer URL with the artifact. If the artifact was returned in a form, then the browser POSTs the artifact to the Assertion Consumer URL.

The following steps reflect the back-channel call that the SP FWS Assertion Consumer service makes to the IdP FWS Artifact Resolution Service to resolve the artifact into a response message.

- a. The SP FWS obtains the artifact from the GET or POST data, depending on how the IdP FWS is configured to redirect the browser. FWS then obtains the SOAP endpoint of the Artifact Resolution Service from the IdP configuration information. The source ID is part of the artifact. After the SOAP endpoint is obtained, the SP FWS makes a back-channel call to the IdP FWS Artifact Resolution service to resolve the artifact.
- b. The IdP FWS requests the response message from the local Policy Server. The message that is stored as a session variable is requested using the Java Agent API. The session ID is extracted from the artifact. The session variable name is the string representation of the artifact message handle.

- c. The local Policy Server retrieves the response message from the session store and deletes it after the artifact retrieval.
- d. The local Policy Server returns the response message to the IdP FWS. The IdP FWS returns the response message to the SP FWS Assertion Consumer Service.

The back-channel call is now complete.

- 23. The SP FWS obtains the response message from the post data. The service then determines the target resource from the configuration and makes an isProtected call to the Policy Server for the target resource.

If the assertion is encrypted, the FWS makes a tunnel call. This call takes the encrypted assertion and returns the assertion in the clear.

- 24. The Policy Server returns the realm OID for the target resource.
- 25. The SP FWS passes the response message to the local Policy Server through a login call. The response message acts as the credentials and the realm OID is obtained from the isProtected call.
- 26. The SAML 2.0 authentication scheme logs the user in using the response message as credentials.
- 27. The local Policy Server returns OK to the SP FWS.
- 28. If a success reply is returned, SP FWS creates an SMSESSION cookie for the SP domain. The service places the cookie in the browser and redirects the user to the target URL, which is obtained from the configuration information.
If the login fails, the SP FWS redirects the user to a No Access URL.
- 29. The browser of the user requests the target URL, which the SP-side Web Agent protects. Because the browser has an SMSESSION cookie, the Web Agent does not challenge the user.

Flow Diagram for SSO Using SAML 2.0 Authentication with POST Binding

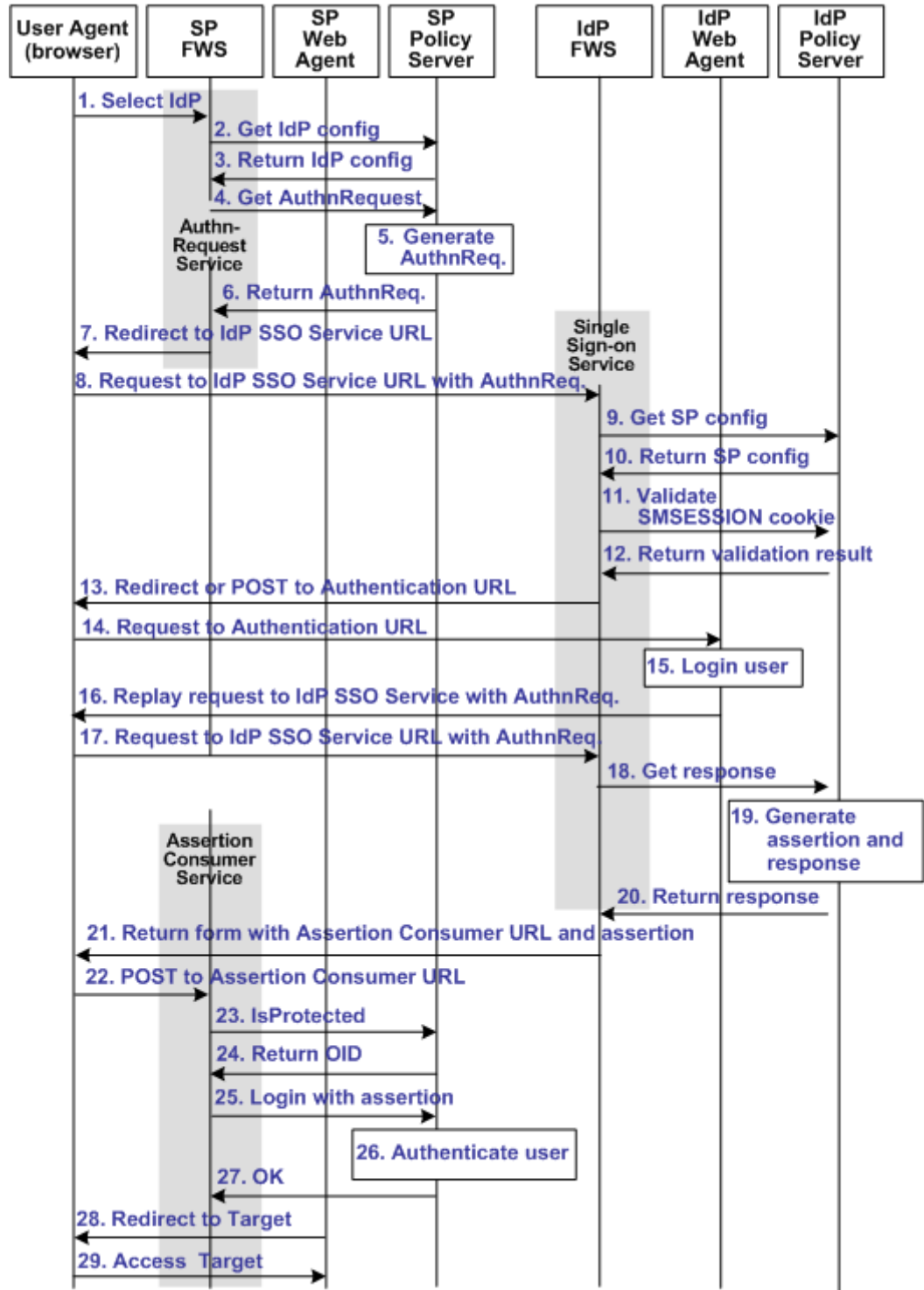
The following illustration shows the detailed flow between a user and the components deployed at an Identity Provider (IdP) and Service Provider (SP) sites. This set-up enables single sign-on between the sites, using SAML 2.0 POST binding as the method of obtaining the SAML assertion for authentication.

The flow diagram assumes the following:

- The SP initiates the request for a resource.
- Successful authentication and authorization at the IdP and SP sites.

Note: This flow applies to examples that do not use the SAML Affiliate Agent.

The flow diagram for SAML 2.0 authentication-POST binding follows.



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

The sequence of events is as follows:

1. The user chooses a link at the SP to authenticate at a specific IdP. This link must include a Provider ID representing the chosen IdP.
2. SP FWS requests the IdP configuration information from the local Policy Server.
3. The Policy Server returns the IdP configuration information to SP FWS. FWS can cache this configuration information.
4. SP FWS requests an AuthnRequest message from the local Policy Server through a tunnel call, passing the Provider ID.
5. The Policy Server generates the AuthnRequest message in an HTTP redirect binding.
6. The local Policy Server returns the AuthnRequest message to the SP FWS in an HTTP redirect binding.
7. SP FWS redirects the user to the IdP Single Sign-on Service URL, which is obtained from the configuration information with the AuthnRequest message.
8. The browser requests the IdP Single Sign-on Service URL.
9. IdP FWS requests the SP configuration information from the IdP local Policy Server.
10. The local Policy Server returns the configuration information.

Note: FWS can cache the configuration information.

11. IdP FWS gets an SMSESSION cookie for this domain of the IdP and calls the Policy Server to validate it. If there is no SMSESSION cookie, the IDP FWS redirects or posts to the Authentication URL.
12. The Policy Server verifies the validity of the SMSESSION cookie and returns the result.
13. If the SMSESSION cookie does not exist or is not valid, the IDP FWS redirects or posts to the Authentication URL. FWS obtains this URL from the configuration information. If the SMSESSION cookie is valid, the IDP FWS skips to 18.
14. The browser requests the Authentication URL, which the IdP Web Agent protects.
15. The IdP Web Agent logs the user in, setting the SMSESSION cookie and lets the request pass to the Authentication URL.
16. The Authentication URL is the redirect.jsp file, which replays the request to the IdP single sign-on service with the AuthnRequest message.
17. The browser requests the IdP single sign-on service URL. This request is equivalent to the request from step 8, but now the user has a valid SMSESSION cookie.

18. The IdP FWS requests a SAML 2.0 assertion from the Policy Server. The AuthnRequest goes through an authorize call to the realm obtained from the configuration information.
19. The Policy Server generates an assertion that is based on the configuration information for the SP, signs it, and returns the assertion wrapped in a response message.
20. The response message is returned to IdP FWS.
21. IdP FWS returns a form to the user. The form contains the response message, the Assertion Consumer URL, obtained from the configuration information, and the JavaScript to submit the form.
Note: If the assertion generator indicates that the current sessions authentication level too low, the IdP FWS redirects to the authentication URL as in Step 13 to facilitate step-up authentication.
22. The browser posts the response message to the Assertion Consumer URL at the SP.
23. The SP FWS obtains the response message from the POST data. FWS then determines the target resource from the configuration and makes an isProtected call to the Policy Server for the target resource.

If the assertion is encrypted, the FWS makes a tunnel call. The call takes the encrypted assertion and returns the assertion in the clear.
24. The Policy Server returns the realm OID for the target resource.
25. The SP FWS passes the response message to the local Policy Server through a login call. FWS uses the response message as credentials and the realm OID obtained from the isProtected call.
26. The SAML 2.0 authentication scheme logs the user in using the response message as credentials.
27. The local Policy Server returns OK to the SP FWS.
28. If a success reply is returned, SP FWS creates an SMSESSION cookie for the SP domain. FWS then places the cookie in the browser and redirects the user to the target URL, which is obtained from the configuration information.

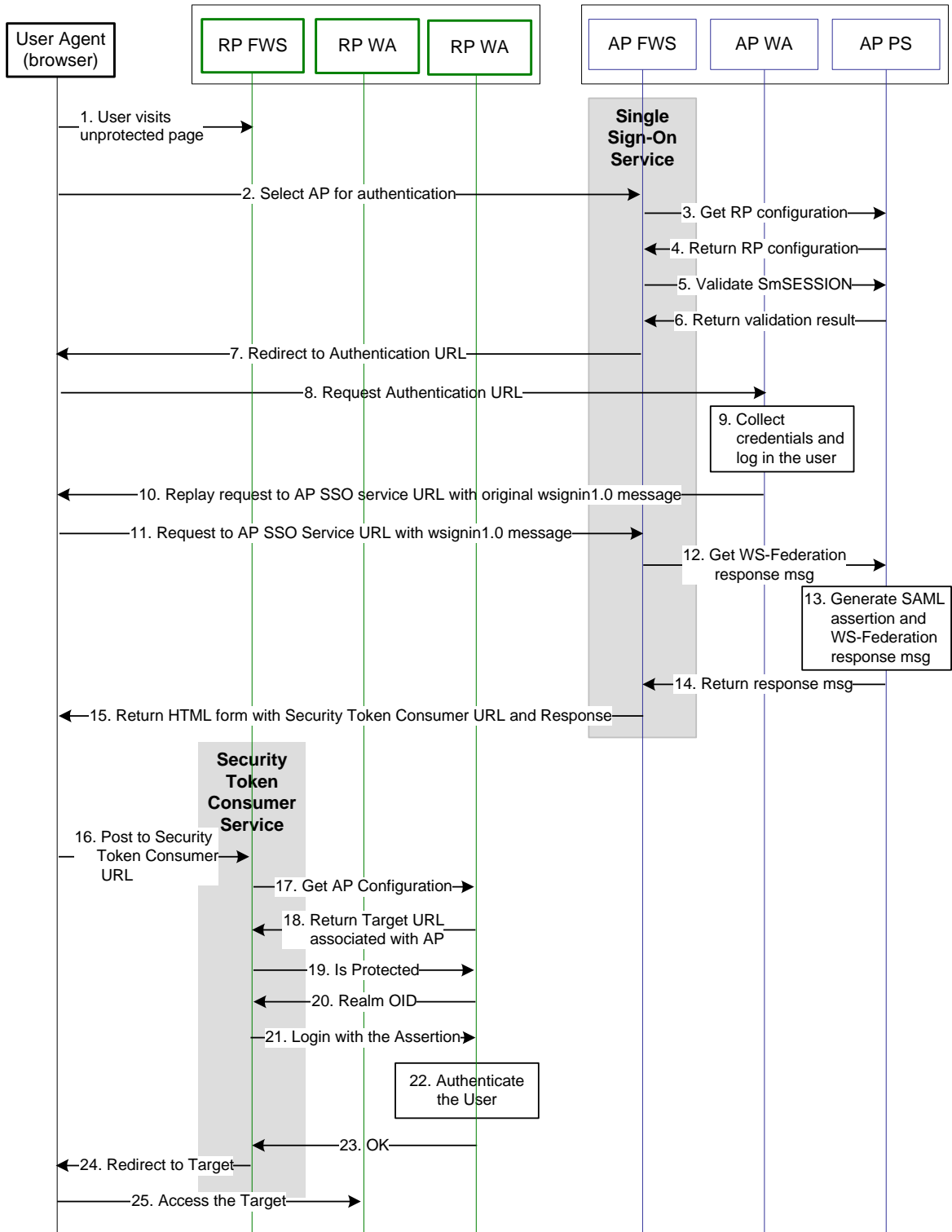
If the login fails, the SP FWS redirects the user to a No Access URL.
29. The browser sends a request to the target URL, which the SP-side Web Agent protects. Because the browser has an SMSESSION cookie, the Web Agent does not challenge the user.

Flow Diagram for WS-Federation SSO Initiated at the Resource Partner

The following illustration shows the detailed flow between a user and the Federation Security Services components at an Account Partner (AP) and Resource Partner (RP) sites. This set-up enables single sign-on between the sites, using WS-Federation as the method of obtaining the SAML assertion for authentication.

The flow diagram assumes the following information

- The Resource Partner initiates the request for a resource.
- Successful authentication and authorization at the AP and RP sites.



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

The WS-Federation single sign-on process is as follows:

1. The user visits an unprotected site selection page at the Resource Partner.
2. The user chooses a link to authenticate for AP that is a federated partner. This link points to the Single Sign-on Service at the AP. The link must contain the Provider ID of the RP and can include some optional parameters, such as the wctx parameter. The browser requests the AP SSO Service URL.
3. Based on RP provider ID specified as a query parameter, the AP FWS requests the RP configuration information from the local Policy Server.
4. The local Policy Server returns the configuration information.

Note: The FWS can cache the configuration information.

5. The AP FWS gets the SMSESSION cookie for the AP domain and calls the Policy Server to validate it. If there is no SMSESSION cookie, the AP FWS skips to step 7.
6. The Policy Server verifies the validity of the SMSESSION cookie and returns the result to the FWS application.
7. If the SMSESSION cookie does not exist or is not valid, the AP FWS redirects the user to the Authentication URL obtained from the RP configuration information. If the SMSESSION cookie is valid, the AP FWS skips to step 12.
8. The browser requests the Authentication URL, which the AP Web Agent protects.
9. The AP WA authenticates the user and sets the SMSESSION cookie. The AP WA lets the request pass to the Authentication URL.
10. The Authentication URL points to the redirect.jsp, which replays the request to the AP SSO service with the original wsignin message.
11. The browser requests the AP SSO Service URL. This request is equivalent to the request from step 2, but now the user has a valid SMSESSION cookie.
12. The AP FWS requests a WS-Federation <RequestSecurityTokenResponse> from the Policy Server through an authorize call to the realm obtained from the configuration information.
13. The Policy Server generates a SAML1.1 assertion that is based on the configuration information for the RP. The Policy Server signs the assertion and returns it wrapped in an <RequestSecurityTokenResponse> message.
14. The <RequestSecurityTokenResponse> message is returned to the AP FWS.

15. The AP FWS returns a form to the user containing the following information:

- URL encoded <RequestSecurityTokenResponse> message.
- Security Token Consumer Service URL.
- Optional wctx that came with the wsignin message.
- JavaScript to auto submit the form.

If the original wsignin request contains the wreply parameter, its value becomes the Security Token Consumer URL. The wreply value becomes the URL only if the Security Token Consumer URL setting is not in the RP configuration information. For security reasons, the Security Token Consumer URL setting in the RP configuration information takes precedence over the wreply parameter.

Note: The assertion generator can indicate that the authentication level of the current session is too low. If the level is too low, the AP FWS redirects to the authentication URL as in step 7 to facilitate “step-up” authentication.

16. The user agent posts the <RequestSecurityTokenResponse> message and wctx to the Security Token Consumer URL at the RP.
17. The RP FWS obtains the <RequestSecurityTokenResponse> message and wctx from the POST data. RP FWS requests the AP configuration information from the local Policy Server.
18. RP FWS determines the target resource from the AP configuration information from local Policy Server. If the target resource is not part of the AP configuration, and the wctx parameter is found in the POST data, the wctx value becomes the target resource.
19. FWS makes an isProtected call to the Policy Server for the target resource.
20. The Policy Server returns the realm OID for the target resource.
21. The RP FWS passes the <RequestSecurityTokenResponse> message to the local Policy Server through a login call. The <RequestSecurityTokenResponse> message and the realm OID obtained from the isProtected call service as credentials.
22. The WS-Federation authentication scheme logs the user in using the <RequestSecurityTokenResponse> message as credentials.
23. The local Policy Server returns an OK status message to the RP FWS.
24. The RP FWS creates the SMSESSION cookie for the RP domain. FWS places the cookie in the browser and redirects the user to the Target URL or to the wctx POST data. If the login fails, the RP FWS redirects the user to a No Access URL.
25. The user agent requests the Target URL that the RP-side Web Agent protects. Because the browser has the SMSESSION cookie for the RP domain, the Web Agent does not have to challenge the user.

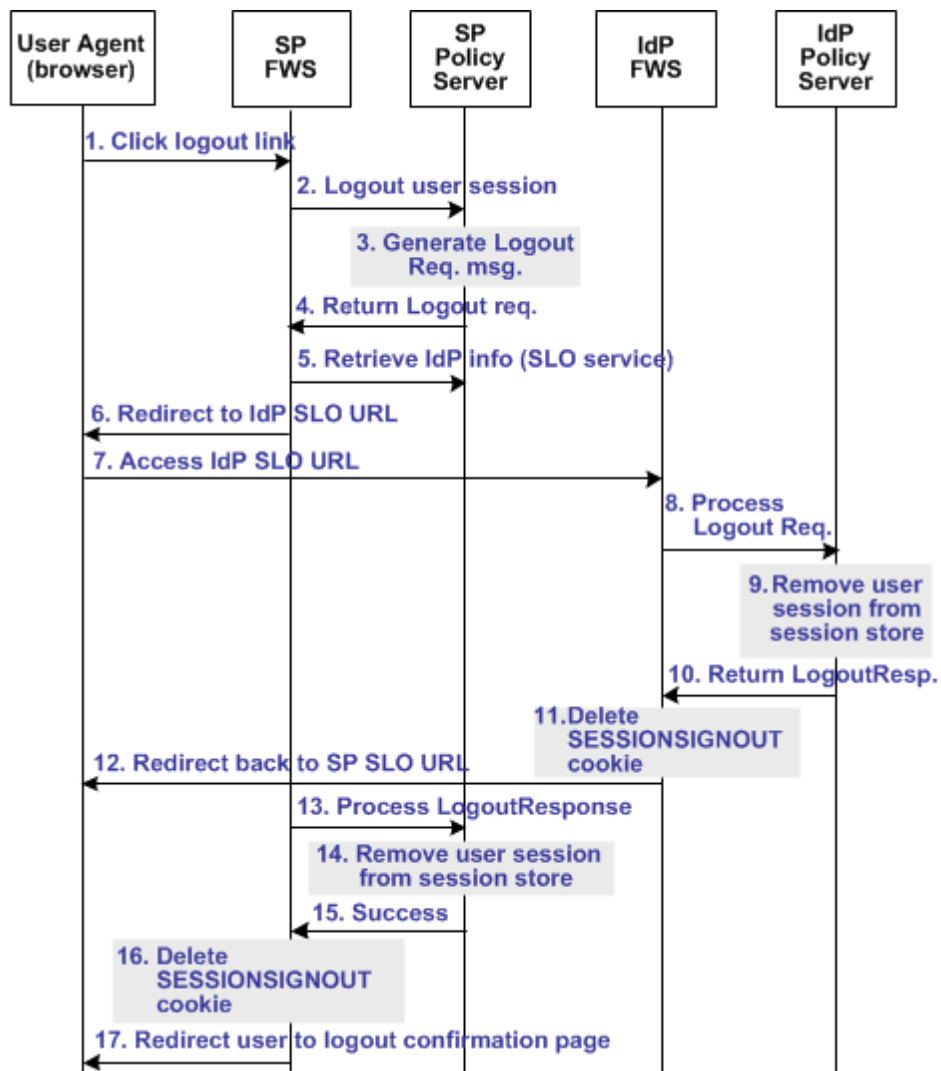
WS-Federation SSO Initiated at the Account Partner

Single sign-on that is initiated by the Account Partner is similar to the RP-initiated use case. HTML content at AP contains intersite transfer links to different RP sites. When the user clicks any link, the web browser requests the AP SSO Service URL. The rest of the processing is same as specified in the RP-initiated use case.

Flow Diagram for SAML 2.0 Single Logout

The following illustration shows the detailed flow for a single logout request between a user and the Federation Security Services components at an Identity Provider (IdP) and Service Provider (SP). This set-up enables single logout for all entities that have a session with a particular user.

The following illustration assumes that the SP initiates the log out request.



Note: The SPS federation gateway can replace the Web Agent and Web Agent Option Pack, which provide the FWS application functions. For more information about installing and configuring the SPS federation gateway, see the *CA SiteMinder Secure Proxy Server Administration Guide*.

The sequence of events is as follows:

1. The user clicks a link at SP to end the global session. The browser of the user accesses the Single Logout servlet at the SP.
SP FWS renames the SMSESSION cookie to SESSIONSIGNOUT to invalidate the current session of the user.
2. FWS reads the SessionId value from the SESSIONSIGNOUT cookie and asks the Policy Server to terminate the user session.

3. Based on the session store information, the user session status is changed to a LogoutInProgress state in the session store. The Policy Server determines that the user session is created based on the SAML assertion received from an IdP. The Policy Server generates a LogoutRequest request to invalidate the user session at the IdP.
4. The Policy Server returns a LogoutRequest request to SP FWS. The Policy Server also returns the Provider ID of the IdP and provider type.
5. SP FWS retrieves the provider configuration data of the IdP, which includes the SLO service URL, from the Policy Server.
6. SP FWS redirects the user to the SLO service at the IdP with the SAML LogoutRequest message added as a query parameter.
7. The browser of the user accesses SLO service at the IdP.

When the IdP FWS receives a LogoutRequest message, it renames the SMSESSION cookie to SESSIONSIGNOUT.
8. The IdP processes the signed LogoutRequest message. The IdP then tries to invalidate the user session at all SPs specified in the session store for that session. The only SP that is not invalidated is the SP that sent the original LogoutRequest.

Note: The process for logging the user out at each SP is similar to Step 2 through Step 7.
9. After terminating the user session from all relevant SPs, the IdP removes the user session from the session store.
10. The IdP Policy Server returns a signed LogoutResponse message to the IdP FWS, containing the provider ID of the SP and provider type. The IdP Policy Server also informs FWS that the user session is removed from the session store.
11. After learning that the user session is removed from the session store, IdP FWS deletes the SESSIONSIGNOUT cookie.
12. The IdP FWS redirects the user to the single logout service at the SP with the SAML LogoutResponse message added as a query parameter. The single logout service is part of the SP FWS application.

The browser of the user accesses SLO service of the SP, which processes the signed LogoutResponse message.

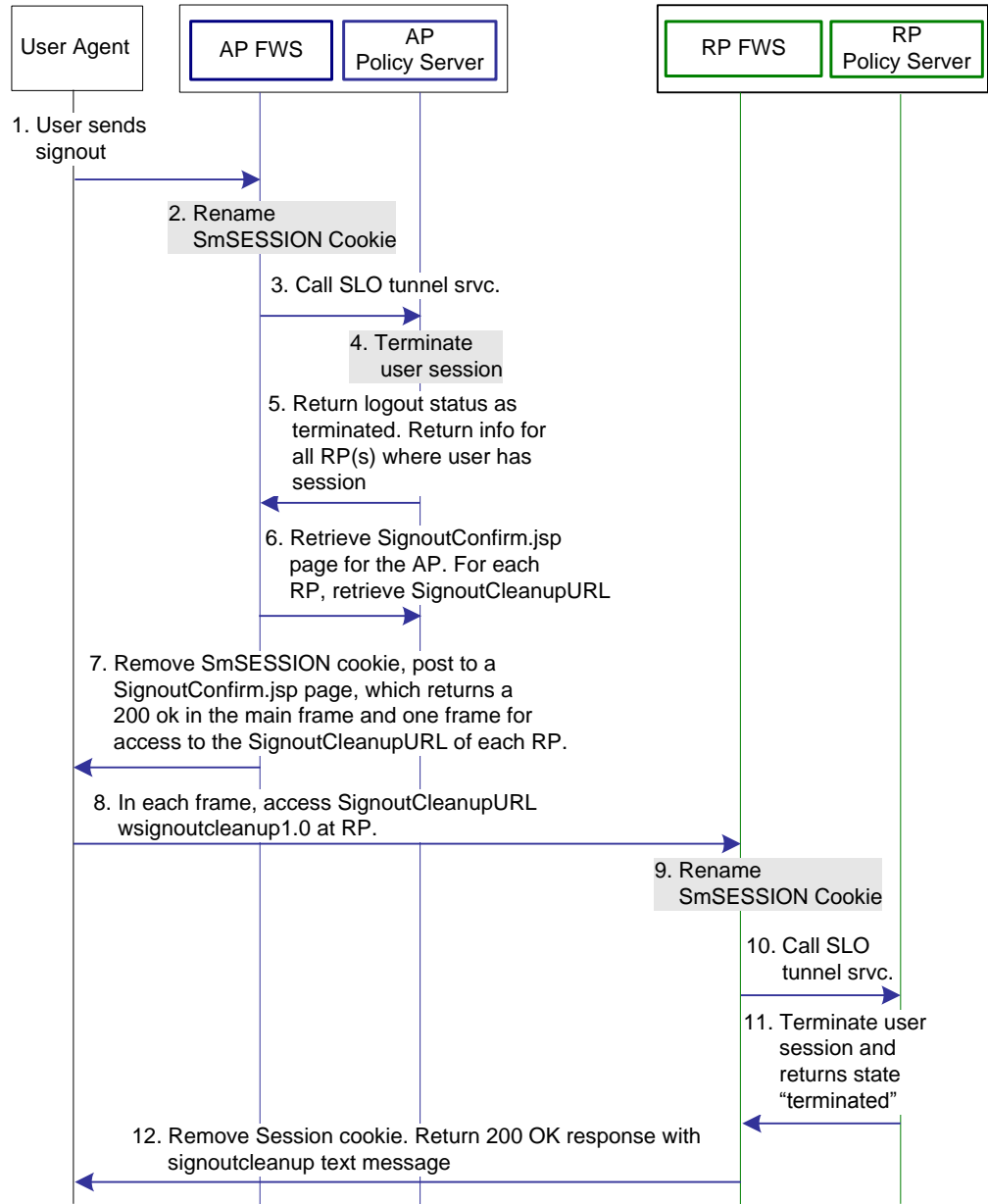
If the LogoutResponse message contains non-SUCCESS return code, FWS issues a SIGNOUTFAILURE cookie, and a base 64-encoded Partner ID is appended to the cookie value. If there are multiple IDs in the cookie, a space character separates them.
13. The SP Policy Server receives the LogoutResponse message from FWS and processes it.
14. The SP Policy Server removes the user session from the session store.

15. After the session is removed from the session store, the Policy Server sends a SUCCESS return code to FWS. The Policy Server includes the SP ID in the final LogoutResponse message.
16. If there are no more LogoutRequest or LogoutResponse messages to process, SP FWS deletes the SESSIONSIGNOUT cookie.
17. FWS redirects the user to the Logout Confirmation page at the SP.

Flow Diagram for WS-Federation Signout (AP-initiated)

The following illustration shows the flow for a signout request between a user and the Federation Security Services components deployed at an Account Partner (AP) and Resource Partner. This set-up enables signout for all entities that have a session with a particular user.

The following illustration assumes that the AP initiates the signout request.



Note: The SPS federation gateway can replace the Web Agent and Web Agent Option Pack, which provide the FWS application functions. For more information about installing and configuring the SPS federation gateway, see the *CA SiteMinder Secure Proxy Server Administration Guide*.

When signout is initiated at the Account Partner, the process flow is as follows:

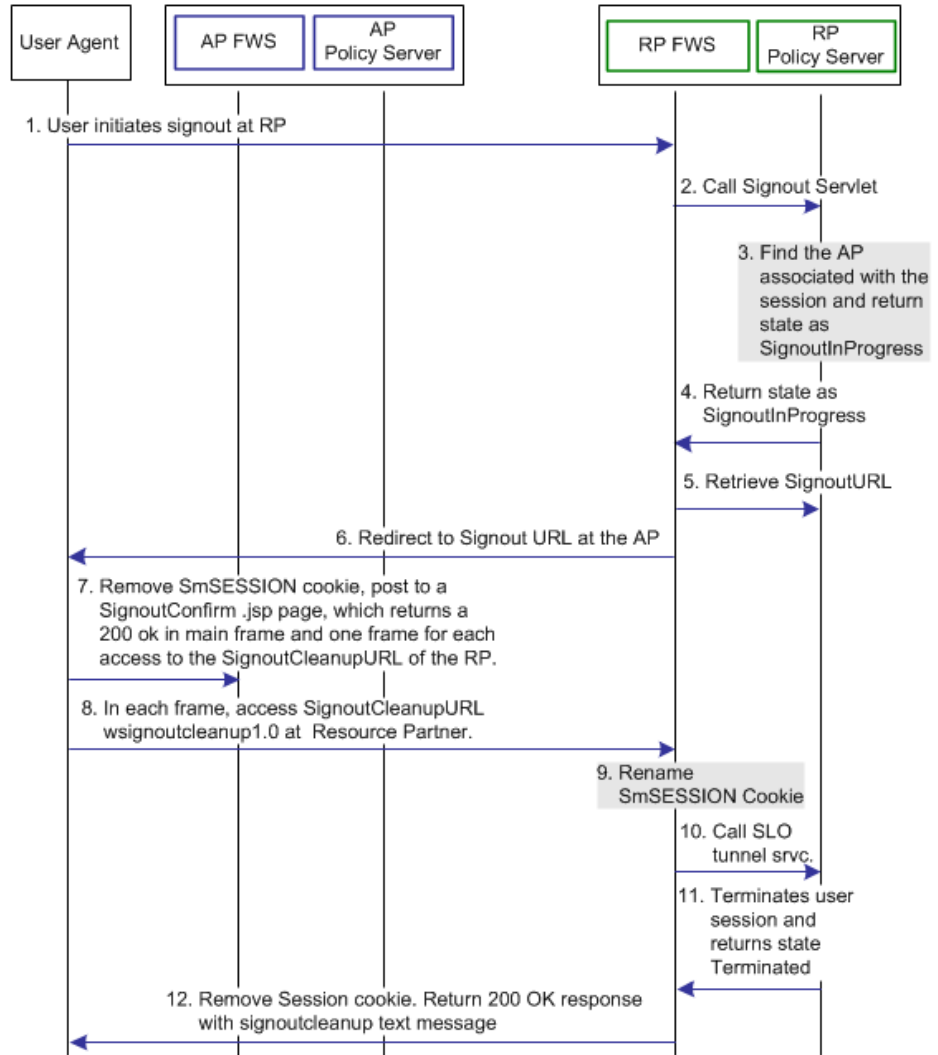
1. The user clicks on a link at the Account Partner to end the global session. The browser of the user sends a HTTP-based wsignout request to the signout servlet at the Account Partner.
2. FWS renames the SMSESSION cookie to SESSIONSIGNOUT to invalidate the current session of the user.
3. FWS reads the SessionId value from the SESSIONSIGNOUT cookie and calls the SLO Tunnel Service API to terminate the user session from the session store.
4. The SLO Tunnel Service API sets the user session status to "Terminated" in the session store. The service also removes all the RP references from the session store that are associated with that user session.
5. The SLO Tunnel Service API returns the logout status "Terminated" to the FWS Signout Servlet. The Tunnel library also returns the RP providerID and providerType for all the RPs associated with the user session.
6. FWS retrieves the provider configuration data of the RP, which includes the signout cleanup URL, from the cache of the provider maintained in FWS.
7. FWS removes the SESSIONSIGNOUT cookie then posts an AP Signout message and multiple RP-SignoutCleanup locations as post data to the SignoutConfirmURL JSP. The SignoutConfirmURL JSP is responsible for parsing various post variables and creating a frame-based HTML page. The mainframe in this HTML page displays the AP-SignOut message. Each of the remaining frames accesses the SignoutCleanupURL of individual RPs associated with the user session.
8. The browser of the user accesses SignoutCleanup service at the Resource Partner site in an individual frame.
9. When the RP FWS (Signout Servlet) receives a wsignoutcleanup request, it renames the SMSESSION cookie to SESSIONSIGNOUT. FWS then calls the SLO Tunnel Service API to process the wsignoutcleanup request.
10. The SLO tunnel library processes the wsignoutcleanup request and terminates the user session from the session store.
11. Then SLO tunnel library returns FWS with a "Terminated" status message indicating that the user session no longer exists in the session store.
12. The FWS Signout Servlet removes the SESSIONSIGNOUT cookie and returns a 200 OK response in the frame.

Note: Steps 8-12 are repeated for individual RPs simultaneously in different frames of the same HTML page.

Flow Diagram for WS-Federation Signout (RP-initiated)

The following illustration shows the flow for a signout request between a user and the Federation Security Services components at an Account Partner (AP) and Resource Partner. This set-up enables signout for all entities that have a session with a particular user.

The following illustration assumes that the RP initiates the sign out request.



Note: The SPS federation gateway can replace the Web Agent and Web Agent Option Pack, which provide the FWS application functions. For more information about installing and configuring the SPS federation gateway, see the *CA SiteMinder Secure Proxy Server Administration Guide*.

When signout is initiated at the Resource Partner, the process flow is as follows:

1. The user clicks a link at the Resource Partner to end the global session. The browser sends a HTTP-based wsignout request to the Signout servlet at the Resource Partner.

Note: The RP site is receiving a wsignout message and not a wsignoutcleanup message.

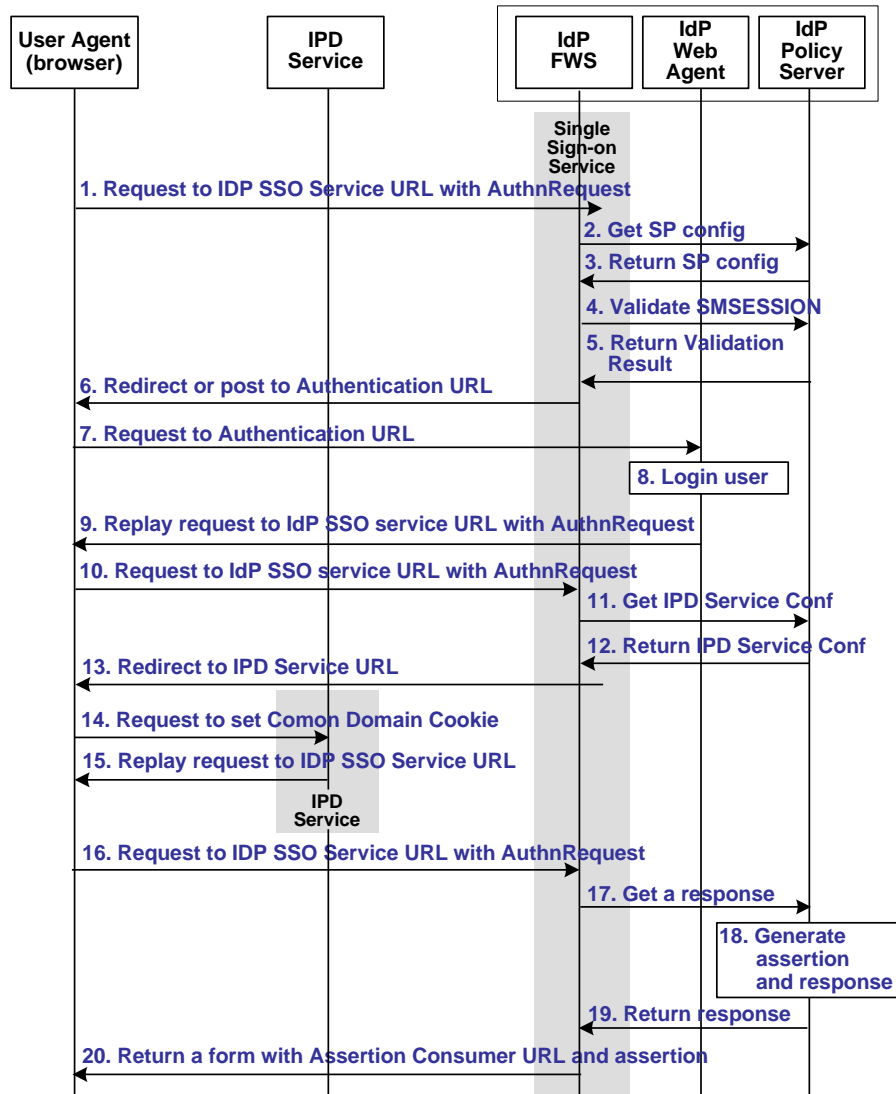
2. FWS reads the SessionId value from the SMSESSION cookie, renames the SMSESSION cookie to SESSIONSIGNOUT, and calls the SLO tunnel library with the wsignout request.
3. Based on information in the session store, the tunnel library determines the AP associated with the user session. The SLO tunnel library sets the user session state to SignoutInProgress, but does not terminate it.
4. The tunnel library returns the SignoutInProgress state message and the Account Partner providerID and providerType.
5. FWS retrieves Account Partner configuration data, which includes the Signout URL, from the FWS cache or Policy Server.
6. FWS redirects the browser of the user to the Signout URL.
7. FWS removes the SESSIONSIGNOUT cookie then posts an AP signout message and multiple RP-SignoutCleanup locations as post data to the SignoutConfirmURL JSP. The SignoutConfirmURL JSP is responsible for parsing various post variables and creating a frame-based HTML page. The primary frame in this HTML page displays the AP-SignOut message. Each of the remaining frames accesses the SignoutCleanupURL of individual RPs associated with the user session.
8. The browser accesses SignoutCleanup service at the Resource Partner site in an individual frame.
9. When RP FWS (Signout Servlet) receives a wsignoutcleanup request, it renames the SMSESSION cookie to SESSIONSIGNOUT. The service then calls the SLO Tunnel Service API to process the wsignoutcleanup request.
10. The SLO tunnel library processes the wsignoutcleanup request and terminates the user session from the session store.
11. Then SLO tunnel library returns FWS with a Terminated status message indicating that the user session no longer exists in the session store.
12. The FWS Signout Servlet removes the SESSIONSIGNOUT cookie and returns a 200 OK response in the frame.

Note: Steps 8-12 are repeated for individual RPs simultaneously in different frames of the same HTML page.

Flow Diagram for Identity Provider Discovery Profile

The following illustration shows the flow for an Identity Provider Discovery service between the user and the Federation Security Services components at an Identity Provider. This set-up involves redirecting from an Identity Provider to the Identity Provider Discovery Profile service to set the common domain cookie.

The following illustration assumes that the SP FWS redirects the user to the IdP SSO Service URL.



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

The Identity Provider Discovery process is as follows:

1. The user agent (browser) requests the IdP SSO Service URL.
2. The IdP FWS requests the SP configuration information from the local Policy Server.
3. The local Policy Server returns the configuration information.

Note: The FWS can cache the configuration information.

4. The IdP FWS gets the SMSESSION cookie for the IdP domain and calls to the Policy Server to validate it. If there is no SMSESSION cookie, the IdP FWS skips to Step 6.
5. The Policy Server verifies the validity of the SMSESSION cookie and returns the result.
6. If the SMSESSION cookie does not exist or is not valid, the IdP FWS redirects or posts to the Authentication URL obtained from the configuration. If the SMSESSION cookie is valid, the IdP FWS skips to Step 18.
7. The user agent requests the Authentication URL. The IdP Web Agent protects the Authentication URL.
8. The IdP Web Agent logs the user in, setting the SMSESSION cookie and lets the request pass to the Authentication URL.
9. The Authentication URL is the redirect.jsp file, which replays the request to the IdP SSO Service with the AuthnRequest message.
10. The user agent requests the IdP SSO Service URL. This request is equivalent to the request from step 8, but now the user has a valid SMSESSION cookie.
11. The IdP FWS requests the Identity Provider Discovery Profile (IPD) configuration from the Policy Server, passing the Identity Provider ID.
12. The Policy Server returns with the IPD configuration, such as IPD Service URL, common domain cookie, and persistence information of the common domain cookie.
13. The IdP FWS redirects the user to the IPD Service URL to set the common domain cookie.
14. The IdP FWS redirects the user to the IPD Service URL.
15. The IPD Service sets or updates the common domain cookie with the Identity Provider ID. The IPD Service redirects the user agent back to the IdP FWS from which it received the Set Request.
16. The user agent requests the IdP SSO Service URL.

17. The IdP FWS requests a SAML 2.0 assertion from the Policy Server, passing the AuthnRequest through an authorize call to the realm obtained from the configuration.
18. The Policy Server generates an assertion that is based on the configuration information for the Service Provider. The Policy Server signs the assertion and returns the assertion wrapped in a response message.
19. The response message is returned to the IdP FWS.
20. The IdP FWS returns a form to the user containing the response message, the Assertion Consumer URL obtained from the configuration and Javascript to submit the form.

Note: The assertion generator can indicate that the authentication level of the current session is too low. If the level is too low, the IdP FWS redirects to the authentication URL as in Step 13 to facilitate step-up authentication.

After the final step in the diagram, the user agent posts the response message to the Assertion Consumer URL at the Service Provider.

Index

A

A Security Issue Regarding SAML 1.x Assertions • 232
Add a CA Certificate for an SSL Back Channel at the SP • 141
Add a Client Certificate to smkeydatabase • 276, 381
Add a Consumer to an Affiliate Domain • 225
Add a Domain Object • 220
Add a Private Key and Certificate to the IdP Smkeydatabase • 145
Add a Public Key to Smkeydatabase at the IdP • 148
Add a Resource Partner to an Affiliate Domain • 403
Add a SAML 2.0 Service Provider to an Affiliate Domain • 287
Add a Web Agent to the Federation Agent Group • 234, 302
Add Entities to an Affiliate Domain • 222
Add Functionality to the Federation Deployment • 134
Add Relying Partners to the FWS Policy for Obtaining Assertions • 303
Add Relying Partners to the FWS Policy for Obtaining Assertions (Artifact SSO) • 235
Add the Service Provider to the Affiliate Domain at the IdP • 113
Add the User Directory to the Affiliate Domain at the IdP • 112
Add Users by Manual Entry for Access to a Service Provider • 290
Add Users by Manual Entry for Resource Partner Access • 408
addCert Option • 199
Adding Users and Groups for Access to a Consumer • 229
Adding Users by Manual Entry • 230
addPrivKey Option • 198
addRevocationInfo Option • 200
Affiliate Domain Overview • 219
AffiliateXMLSignatureImplementation Setting • 194
Affiliation Overview • 343
Affiliations for Single Logout • 344
Affiliations for Single Sign-On • 343
Affwebserver.log and FWSTrace.log Show Wrong Time • 500

Allow Access to Federation Web Services (asserting party) • 158
Allow Access to Federation Web Services (Relying Party) • 166
Allow Nested LDAP Groups Resource Partner Access • 407
Allow Nested LDAP Groups Service Provider Access • 289
Allow the Identity Provider to Assign a Value for the NameID • 310
Allowing Nested Groups Access to Consumers • 230
APIs for Federation Security Services • 29
Artifact Resolution Service (SAML 2.0) • 480
Assertion Consumer Service (SAML 2.0) • 490
Assertion Retrieval Service (SAML 1.x) • 479
Assertion Validity for Single Sign-on • 232, 298
Assign an Administrator • 221
Assign Name IDs to Affiliations • 344
Assign User Directories • 220
Attribute Service (SAML 2.0) • 485
Attribute Types • 244
Attributes for SSO and Attribute Query Requests • 320
Authenticate Users with No SiteMinder Session (SAML 1.x) • 227
Authenticate Users without a SiteMinder Session • 404
Authentication Fails After Modifying Authentication Method • 502
Authentication Users with no SiteMinder Session (SAML 2.0) • 296
AuthnRequest (SAML 2.0) • 489
AuthnRequest Query Parameters Used by a SiteMinder SP • 327
Authorize Users with Attributes from an Assertion Query • 389

B

Backchannel Configuration for HTTP-Artifact SSO • 263
Basic Authentication to Protect the Service that Retrieves Assertions • 237, 315
Basic over SSL to Protect the Assertion Retrieval Service • 237, 315
Bindings for Single Logout • 364

C

- CA Technologies Product References • 3
- Certificate and Private Key Usage for Federation • 183
- Certificate Revocation Lists in the smkeydatabase • 191
- Certificates for SSL Connections • 185
- Certificates Stored in the smkeydatabase Only at the Asserting Party • 190
- Certificates To Secure the Artifact Back Channel • 185
- changePassword Option • 201
- Choosing Whether to Protect the Intersite Transfer URL • 253
- Client Authentication Fails for SAML Artifact Single Sign-on • 502
- Client Certificate Auth to Protect the Service that Retrieves Assertion • 238, 316
- Client Certificate Authentication Across the Back Channel • 185
- Command Options for smfedexport • 453
- Command Options for smfedimport • 459
- Configuration Checklist at the Identity Provider • 286
- Configuration Overview to Supply Attributes as HTTP Headers • 272, 368, 440
- Configuration Settings that Must Use the Same Values • 471
- Configure a Custom SAML 1.x POST Authentication Scheme • 265
- Configure a Custom WS-Federation Auth. Scheme • 445
- Configure a Default or Active Session Model • 242
- Configure a Name ID • 291
- Configure a Name ID for a WS-Federation Assertion • 409
- Configure a Name ID for Inclusion in the Assertion • 116
- Configure a Response to Send Attributes as HTTP Headers • 274, 370, 442
- Configure a SAML 1.x Assertion • 231
- Configure a SAML 2.0 Affiliation (Optional) • 291
- Configure a SAML or WS-Federation Authentication Scheme • 163
- Configure a Shared Session Model • 242
- Configure a Single Target Realm for All Authentication Schemes • 279, 384
- Configure a Single Target Realm for All WS-Federation Authentication Schemes • 445
- Configure a Single Use Policy • 360
- Configure a Unique Realm for Each SAML Authentication Scheme • 278, 383
- Configure a Unique Realm for Each WS-Fed Authentication Scheme • 444
- Configure a Web Agent in the FSS Administrative UI • 86
- Configure Affiliations • 344
- Configure an Affiliate Domain • 219
- Configure an Assertion for One Time Use • 300
- Configure Assertion Attributes for WS-Federation • 417
- Configure Attributes at the Attribute Authority • 395
- Configure Attributes for Assertions (optional) • 319
- Configure Attributes for SAML 1.x Assertions • 245
- Configure Attributes for SSO Assertions • 320
- Configure Attributes for WS-Federation Assertions (optional) • 416
- Configure Attributes to Include in SAML 1.x Assertions (Optional) • 243
- Configure Digital Signing (required for POST Binding) • 145
- Configure Disambiguation Locally as Part of the Authentication Scheme • 355
- Configure ECP at the Identity Provider • 314
- Configure ECP at the Service Provider • 363
- Configure Federation Web Services (Asserting Party) • 157
- Configure Federation Web Services at the Relying Party • 165
- Configure FWS Trace Logging • 465
- Configure Identity Provider Discovery at the IdP • 334
- Configure Indexed Endpoints for the Assertion Consumer Service • 308
- Configure IP Address Restrictions for 1.x Consumers (optional) • 248
- Configure IP Address Restrictions for Service Providers (optional) • 311
- Configure POST Single Sign-on at the IdP • 117
- Configure Request Processing with a Proxy Server • 338
- Configure Request Processing with a Proxy Server at the SP • 379
- Configure Required General Information • 292
- Configure Required General Information for WS-Federation • 409
- Configure SAML 1.x Artifact Authentication • 261

-
- Configure SAML 1.x POST Profile Authentication • 263
 - Configure SAML 2.0 Affiliations At the Identity Provider • 343
 - Configure SAML 2.0 Artifact Single Sign-on • 137
 - Configure SAML 2.0 SSO with Dynamic Account Linking at the SP • 78
 - Configure Signout for WS-Federation • 420
 - Configure Single Logout • 134, 364
 - Configure Single Logout (optional) • 331
 - Configure Single Sign-on at the SP • 357
 - Configure Single Sign-on for SAML 2.0 • 298
 - Configure Single Sign-on for WS-Federation • 411
 - Configure SiteMinder as a Resource Partner • 425
 - Configure SiteMinder as a SAML 1.x Consumer • 255
 - Configure SiteMinder as a SAML 1.x Producer • 223
 - Configure SiteMinder as a SAML 2.0 Identity Provider • 285
 - Configure SiteMinder as a SAML 2.0 Service Provider • 347
 - Configure SiteMinder as an Account Partner • 401
 - Configure SSO with Attributes from a Web Application • 74
 - Configure the Authentication Scheme that Protects the Artifact Service • 236, 314
 - Configure the Back Channel for the Attribute Authority • 395
 - Configure the Backchannel for HTTP-Artifact SSO • 359
 - Configure the Backchannel for the Attribute Query • 398
 - Configure the Client Certificate Authentication at the Relying Party • 381
 - Configure the FWS Properties File • 124
 - Configure the FWS Properties File at the IdP • 109
 - Configure the NameID for the Attribute Query • 398
 - Configure the Rule for the Single Target Realm • 283, 387
 - Configure the SAML 1.x AMAssertionGenerator.properties File • 171
 - Configure the SAML 1.x Artifact Scheme Setup • 261
 - Configure the SAML 2.0 Authentication Scheme • 352
 - Configure the SAML 2.0 Authentication Scheme at the SP • 128
 - Configure the Service Provider • 118
 - Configure the Single Target Realm • 282, 386, 446
 - Configure the Web Server with the Web Agent Option Pack • 106, 122
 - Configure the WS-Federation Authentication Scheme • 428
 - Configure Time Restrictions for 1.x Consumers (optional) • 248
 - Configure Time Restrictions for Service Provider Availability (optional) • 312
 - Configure WS-Federation Single Sign-on at the Resource Partner • 431
 - Configure WS-Federation Authentication Schemes for the Single Target Realm • 445
 - Considerations Before Migrating Key Databases • 210
 - Consumer Not Authenticating When Accessing Assertion Retrieval Service • 502
 - Contact CA Technologies • 3
 - Conventions in the Installation Overview Procedures • 152
 - Cookie Domain Mismatch Errors • 495
 - Create a Custom SAML 2.0 Authentication Scheme (optional) • 354
 - Create a Custom SAML Artifact Authentication Scheme (Optional) • 263
 - Create a Custom WS-Federation Authentication Scheme (optional) • 432
 - Create a Federation Attribute Variable • 399
 - Create a Policy Expression with the Federation Attribute Variable • 399
 - Create a Policy to Implement Attributes as HTTP Headers • 275, 371, 443
 - Create a Policy to Protect the Authentication URL • 227, 297, 405
 - Create a Policy Using the Single Target Realm • 283, 388, 447
 - Create an Authorization Rule to Validate Users • 273, 369, 441
 - Create Links to Initiate Single Sign-on (optional) • 167
 - Create Links to Target Resources (optional) • 158
 - Create SAML Authentication Schemes for the Single Target Realm • 280, 384
 - Create the Custom Authentication Scheme • 280, 385
 - Create the Policy to Protect the Retrieval Service • 239, 317
 - Create the SAML 1.x POST Common Setup and Scheme Setup • 264
 - createDB Option • 198
 - Creating Affiliate Domains • 219

Creating Links to Consumer Resources for Single Sign-on • 251
Customize a SAML Response Element (optional) • 339
Customize a WS-Federation Assertion (optional) • 414
Customize Assertion Processing with the Message Consumer Plug-in • 265, 373, 434
Customize the SAML 1.x Assertion Response (optional) • 249
Customize the Session Duration in the Assertion • 301
Customizing SAML 2.0 Assertion Responses • 24

D

DBLocation Setting • 193
DBUpdateFrequencyMinutes Setting • 195
Debugging Features • 28
Decrypt an Encrypted Assertion at the SP • 149
Define Indexed Endpoints for Different Single Sign-on Bindings • 305
delete Option • 202
deleteDB Option • 201
deleteRevocationInfo Option • 201
Deploy a Message Consumer Plug-in • 267, 375, 436
Deploy Federation Using a Manual Configuration • 95
Deploy Federation Using the Sample Application • 81
Deploy the Assertion Generator Plug-in • 250, 341, 415
Determine Digital Signing Options • 294, 309
Digital Signing Options at the Service Provider • 353
Documentation Changes • 4

E

Enable a Persistent Session to Store Assertions at the IdP • 138
Enable Attribute Queries and Specify Attributes • 397
Enable Client Certificate Authentication for the Back Channel(optional) • 275, 380
Enable Encryption in the Policy Server User Interface at the IdP • 148
Enable Identity Provider Discovery Profile (optional) • 334
Enable Policy Server Trace Logging at the IdP • 105
Enable ServletExec to Write to the IIS File System • 107

Enable Signature Validation at the SP • 147
Enable Signout • 421, 433
Enable Single Logout • 364
Enable Single Logout at the IdP • 134
Enable Single Logout at the SP • 135
Enable SSL for the IdP Web Server for Artifact Single Sign-on • 138
Enable the Artifact Binding for SAML Authentication at the SP • 142
Enable the Assertion Generator Plug-in • 250
Enable the Assertion Generator Plug-in (SAML 2.0) • 341
Enable the Assertion Generator Plug-in (WS-Federation) • 415
Enable the Creation of a Name Identifier • 310
Enable the Message Consumer Plug-in (SAML 2.0) • 376
Enable the Message Consumer Plug-in for SAML 1.x • 268
Enable the Message Consumer Plug-in for WS-Federation • 436
Enable the Session Store • 176
Enable the Signing of SAML POST Responses • 158
Enable Trace Logging for Federation Components at the SP • 121
Enable Web Agent Option Pack Logging at the IdP • 110
Enable Web Agent Option Pack Logging at the SP • 125
Enabling Encryption • 336
Encrypt a NameID and an Assertion • 336
Encrypt and Decrypt the Assertion • 148
Encrypted Private Key Fails to Be Imported into SMkeydatabase • 498
EncryptedPassword Setting • 194
Encryption/Decryption Operation • 185
Enforce Assertion Encryption Requirements for Single Sign-on • 365
Enforce the Authentication Scheme Protection Level for SSO • 309
Enforce the Policies that Protect Federation Web Services • 181
Enforcing a Single Use Policy to Enhance Security • 359
Enhanced Client or Proxy Profile Overview (SAML 2.0) • 312, 362
Environments that Require a Shared Session Store • 177

Error After Successful Authentication at Consumer/SP • 496
Error During Initialization of JVM • 499
Error Message When Viewing FederationWSCustomUserStore • 497
Exclude a User or Group from Service Provider Access • 289
Excluding a User or Group from Access to a Consumer • 229
Excluding a User or Group from Resource Partner Access • 407
Export Metadata Tool • 450
export Option • 202

F

Features Associated with FWS Policies • 180
Federated Single Sign-on with Security Zones • 30
Federation Data Stored in the Session Store • 175
Federation Security Services Overview • 19
Federation Security Services Process Flow • 505
Federation Security Services Trace Logging • 463
Federation Services URLs • 477
Federation Use Cases • 31
Federation Web Services Access • 118
Federation Web Services Application • 24
Federation Web Services Fails to Send SAML Request to Producer/IdP • 497
Federation Web Services URLs Used by SiteMinder • 477
FederationSample.conf Settings • 86
findAlias Option • 203
Flow Diagram for Authorizing a User with User Attributes • 392
Flow Diagram for Identity Provider Discovery Profile • 530
Flow Diagram for SAML 2.0 Single Logout • 522
Flow Diagram for SSO Using SAML 1.x Artifact Authentication • 505
Flow Diagram for SSO Using SAML 1.x POST Profile Authentication • 508
Flow Diagram for SSO Using SAML 2.0 Authentication with Artifact Binding • 510
Flow Diagram for SSO Using SAML 2.0 Authentication with POST Binding • 514
Flow Diagram for WS-Federation Signout (AP-initiated) • 525
Flow Diagram for WS-Federation Signout (RP-initiated) • 528

Flow Diagram for WS-Federation SSO Initiated at the Resource Partner • 518
Flush Federation Web Services Cache for Trace Logs • 470
Form the Policy to Protect the Target Resource • 279, 383
Formats Supported by the Smkeydatabase • 192
FWS Log Messages at the Policy Server • 465
FWS Log Messages at the Web Agent • 463
FWS Template Sample • 468

G

General Issues • 495
Generate an Assertion for One Time Use • 233
Grant Access to Federation Web Services • 179
Grant Access to the Service for Assertion Retrieval (Artifact SSO) • 234, 302
Guidelines for the Single Logout Confirmation Page • 333

H

help Option • 204
How the Single Use Policy is Enforced • 360
How to Configure a SiteMinder Account Partner • 402
How to Configure a SiteMinder Identity Provider • 286
How to Configure a SiteMinder Resource Partner • 426
How to Configure a SiteMinder Service Provider • 351
How to Configure an Attribute Authority and a SAML Requester • 393
How To Configure SiteMinder as a SAML 1.x Consumer • 256
How To Configure SiteMinder to Act as a SAML 1.x Producer • 224
How To Migrate the Key Databases • 211
How To Protect a Resource with a SAML 1.x Authentication Scheme • 277
How To Protect a Target Resource with a WS-Federation Authentication Scheme • 443
How To Protect Resources with a SAML 2.0 Authentication Scheme • 382
How To Run the Sample Application • 85
How to Use the Configuration Settings Tables • 471
HTTP 404 Error When Trying to Retrieve Assertion at the Consumer • 496

HTTP Error Handling at the IdP • 339
HTTP Error Handling for SAML 2.0 Authentication • 378

I

Identify the SP, IdP, and Other General Settings • 116
Identity Provider Data for a Basic Configuration • 98
Identity Provider Data for an Advanced Configuration • 99
Identity Provider Discovery Profile Service (SAML 2.0) • 485
Identity Provider Profiler Sample • 470
Identity Provider-initiated SSO (POST or artifact binding) • 324
IDP Discovery Configuration at the Service Provider • 371
Implement the AssertionGeneratorPlugin Interface • 249, 340, 414
Implement the MessageConsumerPlugin Interface • 266, 374, 435
Implement WS-Federation Signout • 433
Import Metadata Tool • 457
importDefaultCACerts Option • 203
Include an Allow/Create Attribute in Authentication Requests • 361
Include an Attribute in the Assertion • 144
Indexed Endpoints Flow Diagram • 306
Initiate SAML 1.x Single Sign-On at the Producer • 159
Initiate SAML 2.0 Single Sign-On at the Identity Provider • 160
Initiate SAML 2.0 Single Sign-on at the SP (optional) • 167
Initiate Single Sign-on at the Account Partner • 422
Initiate Single Sign-on at the Resource Partner • 423
Initiate WS-Federation Single Sign-on at the Account Partner • 160
Initiate WS-Federation Single Sign-on at the Resource Partner • 168
Install a Web Agent or SPS Federation Gateway (Relying Party) • 164
Install a Web Agent or SPS Federation Gateway at the Asserting Party • 155
Install a Web or Application Server for the Web Agent Option Pack (Relying Party) • 164
Install an Application Server for the Web Agent Option Pack (Asserting Party) • 156

Install and Configure ServletExec to work with FWS at the IdP • 106
Install and Configure ServletExec to Work with FWS at the SP • 123
Install the Asserting Party Web Agent Option Pack • 156
Install the IdP Policy Server • 102
Install the IdP Web Agent • 105
Install the IdP Web Agent Option Pack • 106
Install the JDK for Federation Web Services • 106, 123
Install the Policy Server at the Asserting Party • 154
Install the Relying Party Policy Server • 162
Install the SP Policy Server • 119
Install the SP Web Agent • 122
Install the SP Web Agent Option Pack • 122
Install the Web Agent Option Pack at the Relying Party • 165
Installation Overview • 151
Internationalization in Federation Security Services • 30
Intersite Transfer Service URL (SAML 1.x) • 478
Introduction to SiteMinder Federation Security Services • 19
IXMLEncryptDecryptImplementation Setting • 194
IXMLSignatureImplementation Setting • 194

J

Java Assertion Generator Plugin API • 29
Java Message Consumer Plugin API • 29

L

LDAPAccessTimeout • 195
Legacy Federation Sample Application Overview • 81
Legacy Sample Application Deployment • 81
listCerts Option • 203
listRevocationInfo Option • 203
Locate User Records for Authentication • 429
Look Up User Records for SAML 2.0 Authentication • 354

M

Manual Deployment Prerequisites • 96
Manual SiteMinder-to-SiteMinder Deployment Overview • 95
Matching Parameter Case-Sensitivity Configuration Issues • 497

- Migrate AM.keystore and Update smkeydatabase • 208
- Modify the FederationSample.conf File • 86
- Modify the Key Database Using smkeytool • 195
- Modify the SetupFederationSample.pl Script (Optional) • 88
- Multibyte Characters in Assertions are Not Handled Properly • 498

N

- NativeDBName Setting • 193

O

- Obtain a LoginID for a WS-Federation User • 430
- Obtain the LoginID • 356
- ODBC Errors Deleting Expiry Data From Session Store • 504
- Optional Configuration Tasks at a 1.x Producer • 225
- Optional Configuration Tasks for a SiteMinder Account Partner • 403
- Optional Configuration Tasks for Identifying a Service Provider • 287
- Optional Tasks to Configure a Service Provider • 351
- Optional Tasks to Configure a SiteMinder Consumer • 256
- Optional Tasks to Configure a SiteMinder Resource Partner • 427
- Overview of a SiteMinder Federation Setup • 151

P

- Perform Authorizations with an Attribute Authority • 389
- Permit Access to the FWS Policy that Protects the Artifact Resolution Service • 139
- Permit the Creation of a Name Identifier for SSO • 361
- Point the Policy Server to the IdP LDAP Policy Store • 103
- Point the Policy Server to the SP LDAP Policy Store • 120
- Policies that Protect Federation Web Services • 179
- Policy Management API • 29
- Policy Server System Fails After Logoff • 498
- Prerequisites for a SiteMinder Asserting Party • 223, 285, 401
- Prerequisites to Deploy the Sample Application • 84
- printCert Option • 204

- Processing Import Files with Multiple Certificate Aliases • 461
- Processing Import Files with Multiple SAML 2.0 Providers • 461
- Properties File for the Key Database • 192
- Protect Target Resources at the Relying Party • 163
- Protect the Artifact Resolution Service at the Identity Provider • 382
- Protect the Authentication URL (SAML 2.0) • 113
- Protect the Target Resource at the SP • 129
- Protecting Against Cross-Site Scripting • 217
- Protecting Federated Communication • 215

Q

- Query Parameter Processing by a SiteMinder IdP • 330

R

- Redirect Users After Failed Authentication Attempts • 437
- Redirect Users After Failed SAML 1.x Authentication Attempts • 269
- renameAlias Option • 204
- Request Processing with a Proxy Server at the IdP • 337
- Request Processing with a Proxy Server at the SP • 379
- Resolving Signature Verification Failures • 500
- Review Application-Generated SiteMinder Objects • 94
- Review the JVMOptions File Which Creates a JVM • 172
- Role of the Smkeydatabase at the Asserting Party • 189
- Role of the Smkeydatabase at the Relying Party • 189
- Run the migratekeystore Tool • 212
- Run the Sample Application on the Policy Server System • 89
- Run the smfedexport Tool • 452
- Run the smfedimport Tool • 458

S

- SAML 1.x Artifact and POST Profiles • 25
- SAML 1.x Artifact Authentication Scheme Overview • 258
- SAML 1.x Artifact Profile Single Sign-On Failing • 501
- SAML 1.x Assertion Generator Properties File • 171

-
- SAML 1.x Authentication Scheme Prerequisites • 255, 350, 425
 - SAML 1.x Authentication Schemes • 256
 - SAML 1.x Matching Configuration Settings • 471
 - SAML 1.x POST Profile Authentication Scheme Overview • 260
 - SAML 1.x-Only Issues • 501
 - SAML 2.0 Artifact and POST Profiles • 26
 - SAML 2.0 Matching Configuration Settings • 473
 - SAML 2.0-Only Issues • 503
 - SAML Affiliate Agent • 27
 - SAML and WS-Federation Authentication Schemes • 24
 - SAML Assertion Generator • 22
 - SAML Authentication Request Process • 349
 - SAML Credential Collector (SAML 1.x) • 488
 - SAML Profiles Supported by SiteMinder • 20
 - Sample Application Components • 82
 - Sample Federation Network • 96
 - Secure Proxy Server Federation Gateway • 28
 - Securing a Federated Environment • 215
 - Securing Connections Across the Federated Environment • 216
 - Securing the IdP Discovery Target Against Attacks • 335, 372
 - Security Token Consumer Service (WS-Federation) • 491
 - Select the Artifact Binding at the IdP • 140
 - Select the Client Cert Option for Authentication • 277, 381
 - Select Users for Which Assertions Will Be Generated • 406
 - Select Users For Which Assertions Will Be Generated • 288
 - Select Users for which the IdP Generates Assertions • 115
 - Select Users for Which the Producer Generates Assertions • 228
 - Service Provider Data for a Basic Configuration • 100
 - Service Provider Data for an Advanced Configuration • 101
 - Service Provider Template Sample • 470
 - Service Provider-initiated SSO (POST or artifact binding) • 326
 - Set a Password for SAML Artifact Back Channel Authentication • 293
 - Set the Authentication Scheme Protection Level • 412
 - Set the Path Variable on the Policy Server System • 85
 - Set the Redirect Mode to Store SAML Attributes • 272, 368, 440
 - Set the Skew Time WS-Federation Single Sign-on • 410
 - Set the Sync Interval for Shared Sessions • 242
 - Set up a SAML Requestor to Generate Attribute Queries • 396
 - Set up Affiliate Domains and Add Sites to these Domains • 154
 - Set up an Affiliate Domain at the IdP • 112
 - Set Up Asserting Party Components • 153
 - Set Up Encryption for SSO • 365
 - Set Up Links at the IdP or SP to Initiate Single Sign-on • 323
 - Set Up Links to Initiate WS-Federation Single Sign-on • 422
 - Set Up Relying Party Components • 161
 - Set Up smkeydatabase at the SP for Signature Validation • 146
 - Set up the Attribute Authority • 393
 - Set Up the Identity Provider • 102
 - Set Up the IdP Session Store for Artifact Single Sign-on • 137
 - Set Up the IdP User Store • 104
 - Set Up the Service Provider • 118
 - Set Up the SP User Store • 120
 - Set up the Web Agent System • 90
 - Set up Time Restrictions for Resource Partner Availability (optional) • 413
 - Setting a One Time Use Condition for an Assertion • 215
 - Setting Up Sessions for a SAML Affiliate Agent Consumer (optional) • 241
 - Setup the SAML 1.x Assertion Generator File • 171
 - Set-up the smkeydatabase for Artifact Single Sign-on (optional) • 167
 - SetupFederationSample.pl Script Options (fss) • 88
 - Signing and Encrypting Messages to Secure Federated Transactions • 183
 - Signing and Verification Operations • 184
 - Signout Service at the AP (WS-Federation) • 484
 - Signout Service at the RP (WS-Federation) • 493
 - Simplify Logging with Trace Configuration Templates • 467
 - Single Logout Request Validity • 332
 - Single Logout Service at the IdP (SAML 2.0) • 483
 - Single Logout Service at the SP (SAML 2.0) • 492

- Single Sign On Service (SAML 2.0) • 481
- Single Sign-on Service (WS-Federation) • 482
- SiteMinder Administrative User Interfaces • 79
- SiteMinder as a Service Provider • 347
- SiteMinder Components for Federation Security Services • 22
- SiteMinder SAML 2.0 Metadata Tools Overview • 449
- smfedexport Tool Examples • 455
- smfedimport Tool Examples • 459
- SmKeyDatabase Overview • 186
- Smkeytool Command Syntax and Options • 197
- Smkeytool Examples for UNIX Platforms • 206
- Smkeytool Examples for Windows Platforms • 205
- Solution 1
 - Single Sign-on based on Account Linking • 33
- Solution 1 Using SAML 1.x Artifact Authentication • 34
- Solution 1 Using SAML 1.x POST Profile • 35
- Solution 1 Using SAML 2.0 Artifact Authentication • 36
- Solution 1 Using SAML 2.0 POST Binding • 37
- Solution 1 Using WS-Federation Passive Requestor Profile • 38
- Solution 10
 - Single Sign-on with No Name ID at the IdP • 64
- Solution 11
 - SAML Artifact SSO Using Security Zones • 67
- Solution 12
 - SSO with Attributes from a Web Application • 71
- Solution 13
 - SAML 2.0 SSO with Dynamic Account Linking at the SP • 76
- Solution 2
 - Single Sign-on based on User Attribute Profiles • 41
- Solution 3
 - Single Sign-on with no Local User Account • 43
- Solution 4
 - Extended Networks • 47
- Solution 5
 - Single Logout (SAML 2.0) • 49
- Solution 6
 - WS-Federation Signout • 52
- Solution 7
 - Identity Provider Discovery Profile (SAML 2.0) • 56
- Solution 8
 - Multi-protocol Network • 59

- Solution 9
 - SAML 2.0 User Authorization Based on a User Attribute • 62
- SP Not Authenticating When Accessing Assertion Retrieval Service • 503
- Specify IP Address Restrictions for Resource Partners (optional) • 412
- Specify Name Identifiers for SAML 2.0 Assertions • 290
- Specify Name IDs for WS-Federation Assertions • 409
- Specify Redirect URLs for Failed SAML 2.0 Authentication • 376
- Specify the Maximum Length of Assertion Attributes • 247, 322, 419
- Specify the POST Binding Authentication at the SP • 127
- Specify the User Store for the IdP Policy Server • 110
- Specify the User Store for the SP Policy Server • 126
- Specify Users for Disambiguation for SAML Affiliations • 345
- Storing User Session, Assertion, and Expiry Data • 175
- Supply SAML Attributes as HTTP Headers • 270, 366, 438

T

- Terminology for Partners in a Federation • 20
- Test Artifact Single Sign-on • 143
- Test Federation Web Services • 125
- Test Federation Web Services at the IdP • 109
- Test SAML 2.0 Single Sign-on • 131
- Test Single Logout • 136
- Test Single Logout with the Sample Application • 93
- Test Single Sign-on with the Sample Application • 92
- The JVMOptions.txt File • 172
- The Web.xml File • 494
- Trace Logging • 463
- Trace Logging Templates for FWS • 467
- Trace Logging Templates for the IdP and SP • 469
- Trace Logs Not Appearing for IIS Web Server Using ServletExec • 499
- Troubleshooting • 495

U

- Unsolicited Response Query Parameters Used by a SiteMinder IdP • 325

- Update Federation Web Services Data in the Logs • 467
- URLs for Services at the Asserting Party • 477
- URLs for Services at the Relying Party • 487
- Use a SAML Affiliation to Locate a User Record (Optional) • 355
- Use a Script to Create a New Attribute • 420
- Use a Script to Create A New Response Attribute • 248
- Use a Search Specification to Locate a User • 356
- Use a Search Specification to Locate a WS-Federation User • 431
- Use Case 1
 - Single Sign-on Based on Account Linking • 32
- Use Case 10
 - SAML 2.0 Single Sign-on with No Name ID at the IdP • 63
- Use Case 11
 - SAML Artifact SSO Using Security Zones • 66
- Use Case 12
 - SAML 2.0 SSO Using Attributes from a Web Application • 70
- Use Case 13
 - SSO with Dynamic Account Linking at the SP • 75
- Use Case 2
 - Single Sign-on Based on User Attribute Profiles • 40
- Use Case 3
 - Single Sign-on with No Local User Account • 42
- Use Case 4
 - Extended Networks • 45
- Use Case 5
 - Single Logout • 48
- Use Case 6
 - WS-Federation Signout • 51
- Use Case 7
 - Identity Provider Discovery Profile • 54
- Use Case 8
 - Multi-protocol Support • 58
- Use Case 9
 - SAML 2.0 User Authorization Based on a User Attribute • 61
- Use Case for SAML Attributes As HTTP Headers • 270, 366, 438
- Use SAML 2.0 Provider Metadata To Simplify Configuration • 449
- Use the SiteMinder Profiler to Log Trace Messages • 466
- User Mapping • 21

- Using a Script to Create A New Attribute • 322
- Using Multiple Policy Servers for the Sample Application • 90
- Using Multiple Web Agents for the Sample Application • 91

V

- Validate Signed AuthnRequests and SLO Requests/Responses • 295
- Validate Signout Requests that are Digitally Signed • 421
- validateCert Option • 204
- Verify Basic Protection of the Assertion Retrieval Service • 236, 304
- View a List of Service Providers in an Affiliation • 346
- View Authentication Schemes That Use an Affiliation • 346

W

- Web Agent Option Pack Fails to Initialize Due to Invalid smjavaagent.dll • 495
- WebLogic Configuration Required for Back Channel Authentication • 293
- WSFedDispatcher Service at the AP • 487
- WSFedDispatcher Service at the RP • 494
- WS-Federation • 20
- WS-Federation Assertion Generator • 23
- WS-Federation Authentication Scheme Overview • 427
- WS-Federation Configuration Settings • 474
- WS-Federation Passive Requestor Profile • 27
- WS-Federation SSO Initiated at the Account Partner • 522

X

- XMLDocumentOpsImplementation Setting • 193