

# CA SiteMinder®

## Federation Security Services Guide

r6.0 SP6



Second 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 © 2011 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:

- SiteMinder®
- SiteMinder® SAML Affiliate Agent
- SiteMinder® Policy Server Option Pack
- SiteMinder® Web Agent Option Pack

## Contact CA Technologies

### Contact CA Support

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

- 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

### Provide Feedback

If you have comments or questions about CA Technologies product documentation, you can send a message to [techpubs@ca.com](mailto:techpubs@ca.com).

If you would like 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>.



# Contents

---

|                                                                             |           |
|-----------------------------------------------------------------------------|-----------|
| <b>Chapter 1: Federation Security Services Overview</b>                     | <b>23</b> |
| Introduction to SiteMinder Federation Security Services . . . . .           | 23        |
| Federation Use Cases . . . . .                                              | 24        |
| Use Case 1: Single Sign-on Based on Account Linking . . . . .               | 24        |
| Use Case 2: Single Sign-on Based on User Attribute Profiles . . . . .       | 25        |
| Use Case 3: Single Sign-on with No Local User Account . . . . .             | 26        |
| Use Case 4: Extended Networks . . . . .                                     | 27        |
| Use Case 5: Single Logout . . . . .                                         | 28        |
| Use Case 6: WS-Federation Signout . . . . .                                 | 29        |
| Use Case 7: Identity Provider Discovery Profile . . . . .                   | 30        |
| Use Case 8: Multi-protocol Support . . . . .                                | 31        |
| Use Case 9: SAML 2.0 User Authorization Based on a User Attribute . . . . . | 31        |
| Use Case 10: SAML 2.0 Single Sign-on with No Name ID at the IdP . . . . .   | 32        |
| Use Case 11: SAML Artifact SSO Using Security Zones . . . . .               | 33        |
| Use Case 12: SAML 2.0 SSO Using Attributes from a Web Application . . . . . | 34        |
| Use Case 13: SSO with Dynamic Account Linking at the SP . . . . .           | 36        |
| Solutions for Federation Use Cases . . . . .                                | 36        |
| Solution 1: Single Sign-on based on Account Linking . . . . .               | 37        |
| Solution 2: Single Sign-on based on User Attribute Profiles . . . . .       | 44        |
| Solution 3: Single Sign-on with no Local User Account . . . . .             | 46        |
| Solution 4: Extended Networks . . . . .                                     | 49        |
| Solution 5: Single Logout (SAML 2.0) . . . . .                              | 50        |
| Solution 6: WS-Federation Signout . . . . .                                 | 52        |
| Solution 7: Identity Provider Discovery Profile (SAML 2.0) . . . . .        | 54        |
| Solution 8: Multi-protocol Network . . . . .                                | 56        |
| Solution 9: SAML 2.0 User Authorization Based on a User Attribute . . . . . | 58        |
| Solution 10: Single Sign-on with No User ID at the IdP . . . . .            | 59        |
| Solution 11: SAML Artifact SSO Using Security Zones . . . . .               | 61        |
| Solution 12: SAML 2.0 SSO with Attributes from a Web Application . . . . .  | 64        |
| Solution 13: SAML 2.0 SSO with Dynamic Account Linking at the SP . . . . .  | 68        |
| Federation Security Services Concepts . . . . .                             | 71        |
| Security Assertion Markup Language (SAML) . . . . .                         | 71        |
| WS-Federation . . . . .                                                     | 72        |
| Entities in a Federated Network . . . . .                                   | 72        |
| User Mapping . . . . .                                                      | 73        |
| Federated Single Sign-on with Security Zones . . . . .                      | 74        |
| SAML Profiles Supported by SiteMinder . . . . .                             | 75        |

---

|                                                                                |     |
|--------------------------------------------------------------------------------|-----|
| Benefits of SiteMinder Federation Security Services .....                      | 75  |
| SiteMinder Components for Federation Security Services .....                   | 77  |
| SAML Assertion Generator .....                                                 | 77  |
| WS-Federation Assertion Generator .....                                        | 78  |
| SAML and WS-Federation Authentication Schemes .....                            | 79  |
| Federation Web Services .....                                                  | 79  |
| SAML Affiliate Agent .....                                                     | 82  |
| Secure Proxy Server Federation Gateway .....                                   | 82  |
| Debugging Features .....                                                       | 83  |
| APIs for Federation Security Services .....                                    | 83  |
| Internationalization in Federation Security Services .....                     | 84  |
| Federation Security Services Process Flow .....                                | 85  |
| Flow Diagram for SSO Using SAML 1.x Artifact Authentication .....              | 85  |
| Flow Diagram for SSO Using SAML 1.x POST Profile Authentication .....          | 87  |
| Flow Diagram for SSO Using SAML 2.0 Authentication with Artifact Binding ..... | 89  |
| Flow Diagram for SSO Using SAML 2.0 Authentication with POST Binding .....     | 93  |
| Flow Diagram for WS-Federation SSO Initiated at the Resource Partner .....     | 97  |
| Flow Diagram for SAML 2.0 Single Logout .....                                  | 101 |
| Flow Diagram for WS-Federation Signout (AP-initiated) .....                    | 103 |
| Flow Diagram for WS-Federation Signout (RP-initiated) .....                    | 106 |
| Flow Diagram for Identity Provider Discovery Profile .....                     | 108 |
| Where to Find Federation Security Services Information .....                   | 111 |

## **Chapter 2: Deploying Federation with the FSS Sample Application** **113**

|                                                           |     |
|-----------------------------------------------------------|-----|
| Federation Sample Application Overview .....              | 113 |
| Prerequisites for Using the FSS Sample Application .....  | 115 |
| Sample FSS Network .....                                  | 117 |
| How To Run the Sample Application .....                   | 118 |
| Modify the FederationSample.conf File .....               | 118 |
| FederationSample.conf Settings .....                      | 118 |
| SetupFederationSample.pl Command Options .....            | 119 |
| Deploy the Sample Application on One System .....         | 121 |
| Deploy the Sample Application on Two Systems .....        | 122 |
| Test Single Sign-on with the FSS Sample Application ..... | 123 |
| Test Single Logout with the FSS Sample Application .....  | 124 |
| Review Application-Generated SiteMinder Objects .....     | 125 |

## **Chapter 3: Deploying Federation without the FSS Sample Application** **127**

|                                             |     |
|---------------------------------------------|-----|
| Manual FSS-to-FSS Deployment Overview ..... | 127 |
| Manual Deployment Prerequisites .....       | 128 |
| Sample Federation Network .....             | 128 |

---

|                                                                      |     |
|----------------------------------------------------------------------|-----|
| Identity Provider Data for a Basic Configuration .....               | 129 |
| Identity Provider Data for an Advanced Configuration .....           | 131 |
| Service Provider Data for a Basic Configuration .....                | 132 |
| Service Provider Data for an Advanced Configuration .....            | 133 |
| Set Up the Identity Provider .....                                   | 133 |
| Install the IdP Policy Server .....                                  | 134 |
| Point the Policy Server to the IdP LDAP Policy Store .....           | 134 |
| Set Up the IdP User Store .....                                      | 135 |
| Install the IdP Policy Server Option Pack .....                      | 136 |
| Enable Policy Server Trace Logging at the IdP .....                  | 136 |
| Install the IdP Web Agent .....                                      | 137 |
| Install the IdP Web Agent Option Pack .....                          | 137 |
| Configure the Web Server with the Web Agent Option Pack .....        | 137 |
| Enable Web Agent Option Pack Logging at the IdP .....                | 141 |
| Specify the User Store for the IdP Policy Server .....               | 142 |
| Set up an Affiliate Domain at the IdP .....                          | 143 |
| Add the Service Provider to the Affiliate Domain at the IdP .....    | 143 |
| Select Users For Which Assertions Will Be Generated at the IdP ..... | 144 |
| Configure a Name ID for Inclusion in the Assertion .....             | 145 |
| Identify the SP, IdP, and Other General Settings .....               | 146 |
| Configure POST Single Sign-on at the IdP .....                       | 146 |
| Protect the Authentication URL (SAML 2.0) .....                      | 147 |
| Federation Web Services Access .....                                 | 149 |
| Configure the Service Provider .....                                 | 149 |
| Set Up the Service Provider .....                                    | 149 |
| Install the SP Policy Server .....                                   | 150 |
| Point the Policy Server to the SP LDAP Policy Store .....            | 150 |
| Set Up the SP User Store .....                                       | 151 |
| Install the SP Policy Server Option Pack .....                       | 152 |
| Enable Trace Logging for Federation Components at the SP .....       | 152 |
| Install the SP Web Agent .....                                       | 153 |
| Install the SP Web Agent Option Pack .....                           | 153 |
| Configure the Web Server with the Web Agent Option Pack .....        | 153 |
| Specify the POST Binding Authentication at the SP .....              | 156 |
| Configure the SAML 2.0 Authentication Scheme at the SP .....         | 157 |
| Protect the Target Resource at the SP .....                          | 158 |
| Test SAML 2.0 Single Sign-on .....                                   | 160 |
| Add Functionality to the Federation Deployment .....                 | 163 |
| Configure Single Logout .....                                        | 163 |
| Configure SAML 2.0 Artifact Single Sign-on .....                     | 166 |
| Include an Attribute in the Assertion .....                          | 173 |
| Configure Digital Signing (required for POST Binding) .....          | 174 |

---

|                                                                                          |            |
|------------------------------------------------------------------------------------------|------------|
| Encrypt and Decrypt the Assertion .....                                                  | 177        |
| <b>Chapter 4: Overview of a SiteMinder Federation Partnership Setup</b>                  | <b>181</b> |
| Installation Overview .....                                                              | 181        |
| Conventions in the Installation Overview Procedures .....                                | 182        |
| Set Up Producing Authority Components .....                                              | 183        |
| Install the Producing-side Policy Server .....                                           | 184        |
| Install the Producing-side Policy Server Option Pack .....                               | 184        |
| Set up Affiliate Domains and Add Sites to these Domains .....                            | 185        |
| Install a Web Agent or SPS Federation Gateway (Producing-side) .....                     | 186        |
| Install a Web or Application Server for the Web Agent Option Pack (Producing-side) ..... | 186        |
| Install the Producing-side Web Agent Option Pack .....                                   | 187        |
| Configure Federation Web Services (Producing-side) .....                                 | 187        |
| Protect Federation Web Services (Producing-side) .....                                   | 188        |
| Set up a Key Database for Signing POST Responses .....                                   | 190        |
| Create Links to Target Resources (optional) .....                                        | 190        |
| Set Up Consuming Authority Components .....                                              | 193        |
| Install the Consuming-side Policy Server .....                                           | 195        |
| Install the Consuming-side Policy Server Option Pack .....                               | 195        |
| Configure a SAML or WS-Federation Authentication Scheme .....                            | 196        |
| Protect Target Resources (Consuming-side) .....                                          | 196        |
| Install a Web Agent or SPS Federation Gateway (Consuming-side) .....                     | 197        |
| Install a Web or Application Server for the Web Agent Option Pack (Consuming-side) ..... | 197        |
| Install the Consuming-side Web Agent Option Pack .....                                   | 198        |
| Configure Federation Web Services (Consuming-side) .....                                 | 198        |
| Protect Federation Web Services (Consuming-side) .....                                   | 199        |
| Set-up the smkeydatabase for Artifact Single Sign-on (optional) .....                    | 200        |
| Create Links to Initiate Single Sign-on (optional) .....                                 | 200        |
| <b>Chapter 5: Setup the SAML 1.x Assertion Generator File</b>                            | <b>203</b> |
| SAML 1.x Assertion Generator Properties File .....                                       | 203        |
| Configure the SAML 1.x AMAssertionGenerator.properties File .....                        | 203        |
| <b>Chapter 6: Review the JVMOptions File Used to Create a JVM</b>                        | <b>205</b> |
| The JVMOptions.txt File .....                                                            | 205        |
| <b>Chapter 7: Storing User Session, Assertion, and Expiry Data</b>                       | <b>207</b> |
| Federation Data Stored in the Session Server .....                                       | 207        |
| Configure and Enable the Session Server .....                                            | 208        |



---

|                                                        |     |
|--------------------------------------------------------|-----|
| Environments that Require a Shared Session Store ..... | 208 |
|--------------------------------------------------------|-----|

## **Chapter 8: Federation Web Services Application Setup 211**

|                                                                                 |     |
|---------------------------------------------------------------------------------|-----|
| Federation Web Services Application Overview .....                              | 211 |
| Federation Web Services Deployment Descriptors .....                            | 213 |
| Configure the Federation Web Services Properties Files .....                    | 213 |
| Configure the AffWebServices.properties File .....                              | 214 |
| Set up the LoggerConfig.properties File .....                                   | 215 |
| Deploy Federation Web Services as a Web Application .....                       | 217 |
| Configure ServletExec to Work with Federation Web Services .....                | 218 |
| Configure WebLogic to Work with Federation Web Services .....                   | 222 |
| Configure WebSphere to Work with Federation Web Services .....                  | 226 |
| Configure JBOSS or Tomcat to Work with Federation Web Services .....            | 231 |
| Protect the Federation Web Services Application .....                           | 234 |
| Enforce Policies that Protect Federation Web Services .....                     | 235 |
| Protect the Assertion Retrieval or Artifact Resolution Service (optional) ..... | 236 |
| Flush Federation Web Services Cache for Trace Logs .....                        | 238 |

## **Chapter 9: Creating Affiliate Domains 239**

|                                           |     |
|-------------------------------------------|-----|
| Affiliate Domain Overview .....           | 239 |
| Configure an Affiliate Domain .....       | 240 |
| Add a Domain Object .....                 | 240 |
| Assign User Directories .....             | 241 |
| Assign an Administrator .....             | 241 |
| Add Entities to an Affiliate Domain ..... | 242 |

## **Chapter 10: Configure SiteMinder as a SAML 1.x Producer 243**

|                                                           |     |
|-----------------------------------------------------------|-----|
| Prerequisites for Producing SAML 1.x Assertions .....     | 243 |
| Configuration Checklist for 1.x Producer .....            | 244 |
| Required Configuration Tasks at a 1.x Producer .....      | 245 |
| Optional Configuration Tasks at a 1.x Producer .....      | 245 |
| Add a Consumer to an Affiliate Domain .....               | 245 |
| Select Users for Which Assertions Will Be Generated ..... | 247 |
| Adding Users and Groups for Access to a Consumer .....    | 248 |
| Excluding a User or Group from Access to a Consumer ..... | 248 |
| Allowing Nested Groups Access to Consumers .....          | 249 |
| Adding Users by Manual Entry .....                        | 249 |
| Configure SAML 1.x Assertions to Authenticate Users ..... | 250 |
| A Security Issue Regarding SAML 1.x Assertions .....      | 251 |
| Configuring a SAML 1.x Assertion .....                    | 251 |

---

|                                                                                                     |     |
|-----------------------------------------------------------------------------------------------------|-----|
| Create Links to Consumer Resources for Single Sign-on . . . . .                                     | 254 |
| Choosing Whether to Protect the Intersite Transfer URL . . . . .                                    | 255 |
| Allow Access to the Federation Web Services Application . . . . .                                   | 256 |
| Set Up Sessions for a SAML Affiliate Agent Consumer (optional) . . . . .                            | 257 |
| Configure a Default or Active Session Model . . . . .                                               | 257 |
| Configure a Shared Session Model . . . . .                                                          | 258 |
| Configure Attributes to Include in SAML 1.x Assertions (Optional) . . . . .                         | 259 |
| Attribute Types . . . . .                                                                           | 259 |
| Configure IP Address Restrictions for 1.x Consumers (optional) . . . . .                            | 262 |
| Configure Time Restrictions for 1.x Consumers (optional) . . . . .                                  | 263 |
| Customize SAML 1.x Assertion Content (optional) . . . . .                                           | 263 |
| Integrate the Assertion Generator Plug-in with SiteMinder (SAML 1.x) . . . . .                      | 264 |
| Protect the Authentication URL to Create a SiteMinder Session (SAML 1.x) . . . . .                  | 265 |
| Create a Policy to Protect the Authentication URL . . . . .                                         | 266 |
| Protect the Assertion Retrieval Service with Client Certificate Authentication (optional) . . . . . | 267 |
| Use of Client Cert. Auth. with an IIS 5.0 Web Server . . . . .                                      | 268 |
| Create the Assertion Retrieval Service Policy . . . . .                                             | 268 |

## **Chapter 11: Configure SiteMinder as a SAML 1.x Consumer 271**

|                                                                                       |     |
|---------------------------------------------------------------------------------------|-----|
| SAML 1.x Authentication Schemes . . . . .                                             | 271 |
| SAML 1.x POST Profile Authentication Scheme Overview . . . . .                        | 273 |
| SAML 1.x Artifact Authentication Scheme Overview . . . . .                            | 274 |
| Configuration Tasks for SAML 1.x Authentication . . . . .                             | 276 |
| Access the Assertion Retrieval Service with a Client Certificate (optional) . . . . . | 276 |
| Configure the Client Certificate Option at the Consumer . . . . .                     | 277 |
| Protect the Assertion Retrieval Service at the Producer . . . . .                     | 278 |
| SAML 1.x Authentication Scheme Prerequisites . . . . .                                | 279 |
| Install the Policy Server and Its Option Pack . . . . .                               | 279 |
| Install Federation Web Services at the Producer and Consumer . . . . .                | 279 |
| Set Up a Key Database to Sign and Verify SAML POST Responses . . . . .                | 280 |
| Configure SAML 1.x Artifact Authentication . . . . .                                  | 280 |
| Configure the SAML 1.x Artifact Scheme Setup . . . . .                                | 280 |
| Create a Custom SAML Artifact Authentication Scheme (Optional) . . . . .              | 282 |
| Configure SAML 1.x POST Profile Authentication . . . . .                              | 282 |
| Create the SAML 1.x POST Common Setup and Scheme Setup . . . . .                      | 282 |
| Configure a Custom SAML 1.x POST Authentication Scheme . . . . .                      | 283 |
| Configure SAML 1.x User Disambiguation . . . . .                                      | 284 |
| Obtain the LoginID . . . . .                                                          | 284 |
| Specify a Search Specification to Locate a User (SAML 1.x) . . . . .                  | 285 |
| Customize Assertion Processing with the Message Consumer Plug-in . . . . .            | 286 |
| Configure the SAML 1.x Message Consumer Plug-in . . . . .                             | 287 |

---

|                                                                          |     |
|--------------------------------------------------------------------------|-----|
| Integrate the Message Consumer Plug-in for SAML 1.x Authentication ..... | 287 |
| Supply SAML Attributes as HTTP Headers .....                             | 288 |
| Use Case for SAML Attributes As HTTP Headers .....                       | 289 |
| Configuration Overview to Supply Attributes as HTTP Headers .....        | 290 |
| Set the Redirect Mode to Store SAML Attributes .....                     | 291 |
| Create an Authorization Rule to Validate Users .....                     | 291 |
| Configure a Response to Send Attributes as HTTP Headers .....            | 292 |
| Create a Policy to Implement Attributes as HTTP Headers .....            | 293 |
| Specify Redirect URLs for Failed SAML 1.x Authentication .....           | 293 |
| How To Protect a Resource with a SAML 1.x Authentication Scheme .....    | 295 |
| How To Assign a SAML 1.x Authentication Scheme to a Realm .....          | 295 |
| Create a Rule for the Realm .....                                        | 299 |
| Create a Response Associated with the Rule .....                         | 300 |
| Form the Policy to Protect the Target Resource .....                     | 300 |

## **Chapter 12: Configure SiteMinder as a SAML 2.0 Identity Provider 301**

|                                                                                |     |
|--------------------------------------------------------------------------------|-----|
| Configuration Checklist at the Identity Provider .....                         | 301 |
| Required Configuration Tasks to Identify a Service Provider .....              | 302 |
| Optional Configuration Tasks for Identifying a Service Provider .....          | 302 |
| Add a SAML 2.0 Service Provider to an Affiliate Domain .....                   | 303 |
| Select Users For Which Assertions Will Be Generated .....                      | 303 |
| Exclude a User or Group from Service Provider Access .....                     | 304 |
| Allow Nested LDAP Groups Service Provider Access .....                         | 305 |
| Add Users by Manual Entry for Access to a Service Provider .....               | 305 |
| Specify Name Identifiers for SAML 2.0 Assertions .....                         | 306 |
| Configure a Name ID .....                                                      | 306 |
| Configure a SAML 2.0 Affiliation (Optional) .....                              | 307 |
| Configure Required General Information .....                                   | 308 |
| Set a Password for SAML Artifact Back Channel Authentication .....             | 309 |
| Validate Signed AuthnRequests and SLO Requests/Responses .....                 | 309 |
| Configure Single Sign-on for SAML 2.0 .....                                    | 311 |
| Assertion Validity for Single Sign-on .....                                    | 311 |
| Customize the Session Duration in the Assertion .....                          | 313 |
| Define Indexed Endpoints for the Assertion Consumer Service .....              | 315 |
| Indexed Endpoints Flow Diagram .....                                           | 316 |
| Define Indexed Endpoints for Different Single Sign-on Bindings .....           | 318 |
| Enforce the Authentication Scheme Protection Level for SSO .....               | 319 |
| Allow the Identity Provider to Assign a Value for the NameID .....             | 319 |
| Configure IP Address Restrictions for Service Providers (optional) .....       | 320 |
| Configure Time Restrictions for Service Provider Availability (optional) ..... | 321 |
| Allow Access to the Federation Web Services Application .....                  | 322 |

---

|                                                                                                 |     |
|-------------------------------------------------------------------------------------------------|-----|
| Set Up Links at the IdP or SP to Initiate Single Sign-on .....                                  | 323 |
| Identity Provider-initiated SSO (POST or artifact binding) .....                                | 323 |
| Service Provider-initiated SSO (POST or artifact binding) .....                                 | 325 |
| Configure Attributes for Inclusion in Assertions (optional) .....                               | 329 |
| Attributes that Function for SSO and Attribute Query Requests .....                             | 330 |
| Configure Attributes for SSO Assertions .....                                                   | 331 |
| Configure Single Logout (optional) .....                                                        | 333 |
| Single Logout Request Validity .....                                                            | 334 |
| Guidelines for the Single Logout Confirmation Page .....                                        | 334 |
| Enable Identity Provider Discovery Profile (optional) .....                                     | 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) .....                                              | 340 |
| Integrate the Assertion Generator Plug-in with SiteMinder (SAML 2.0/WS-Federation) .....        | 341 |
| Protect the Authentication URL to Create a SiteMinder Session (SAML 2.0) .....                  | 343 |
| Protect the Artifact Resolution Service with Client Certificate Authentication (optional) ..... | 345 |
| Create the Artifact Resolution Service Policy .....                                             | 346 |

## **Chapter 13: Configure SAML 2.0 Affiliations At the Identity Provider 349**

|                                                              |     |
|--------------------------------------------------------------|-----|
| Affiliation Overview .....                                   | 349 |
| Affiliations for Single Sign-On .....                        | 349 |
| Affiliations for Single Logout .....                         | 350 |
| Configure Affiliations .....                                 | 350 |
| Assign Name IDs to Affiliations .....                        | 350 |
| Specify Users for Disambiguation for SAML Affiliations ..... | 351 |
| View a List of Service Providers in an Affiliation .....     | 352 |
| View Authentication Schemes That Use an Affiliation .....    | 352 |

## **Chapter 14: Configure SiteMinder as a SAML 2.0 Service Provider 353**

|                                                                    |     |
|--------------------------------------------------------------------|-----|
| SAML 2.0 Authentication Scheme Overview .....                      | 353 |
| SAML Authentication Request Process .....                          | 355 |
| Configuration Tasks for SAML 2.0 Authentication .....              | 356 |
| SAML 2.0 Authentication Scheme Prerequisites .....                 | 357 |
| Install the Policy Server and Its Option Pack .....                | 357 |
| Install the Web Agent or SPS Federation Gateway .....              | 357 |
| Set Up a Key Database to Sign and Verify SAML POST Responses ..... | 358 |
| Configure the SAML 2.0 Authentication Scheme .....                 | 359 |
| Create a Custom SAML 2.0 Authentication Scheme (optional) .....    | 360 |

---

|                                                                                   |     |
|-----------------------------------------------------------------------------------|-----|
| Configure User Disambiguation for User Look Ups .....                             | 360 |
| Use a SAML Affiliation to Locate a User Record (Optional) .....                   | 361 |
| Configure Disambiguation Locally as Part of the Authentication Scheme .....       | 362 |
| Configure Single Sign-on at the SP .....                                          | 364 |
| Configure the Backchannel for HTTP-Artifact SSO .....                             | 365 |
| Enforcing a Single Use Policy to Enhance Security .....                           | 366 |
| Permit the Creation of a Name Identifier for SSO .....                            | 368 |
| Enable the Enhanced Client or Proxy Profile .....                                 | 369 |
| Supply SAML Attributes as HTTP Headers .....                                      | 370 |
| Use Case for SAML Attributes As HTTP Headers .....                                | 371 |
| Configuration Overview to Supply Attributes as HTTP Headers .....                 | 372 |
| Set the Redirect Mode to Store SAML Attributes .....                              | 373 |
| Create an Authorization Rule to Validate Users .....                              | 373 |
| Configure a Response to Send Attributes as HTTP Headers .....                     | 374 |
| Create a Policy to Implement Attributes as HTTP Headers .....                     | 375 |
| Request Processing with a Proxy Server at the SP .....                            | 376 |
| Configure Request Processing with a Proxy Server at the SP .....                  | 376 |
| Enable Single Logout .....                                                        | 377 |
| Bindings for Single Logout .....                                                  | 377 |
| Configure Single Logout .....                                                     | 378 |
| How To Protect Resources with a SAML 2.0 Authentication Scheme .....              | 378 |
| Assign a SAML 2.0 Authentication Scheme to a Realm .....                          | 379 |
| Create a Rule for the Target Resource .....                                       | 383 |
| Create a Policy for the Target Resource .....                                     | 384 |
| Enforce Assertion Encryption Requirements for Single Sign-on .....                | 384 |
| Set Up Encryption for SSO .....                                                   | 384 |
| Customize Assertion Processing with the Message Consumer Plug-in .....            | 385 |
| Configuring the Message Consumer Plug-in (SAML 2.0) .....                         | 386 |
| Specify Redirect URLs for Failed SAML 2.0 Authentication .....                    | 388 |
| Access the Artifact Resolution Service with a Client Certificate (optional) ..... | 390 |
| Configuring the Client Certificate Option at the Service Provider .....           | 390 |
| Select the Client Cert Option for Authentication .....                            | 390 |
| Add a Client Certificate to the SMKeyDatabase .....                               | 391 |
| Protect the Artifact Resolution Service at the Identity Provider .....            | 391 |

## Chapter 15: Use an Attribute Authority to Authorize Users 393

|                                                                |     |
|----------------------------------------------------------------|-----|
| Perform Authorizations with an Attribute Authority .....       | 393 |
| Flow Diagram for Authorizing a User with User Attributes ..... | 396 |
| Configure an Attribute Authority and a SAML Requester .....    | 397 |
| Set up the Attribute Authority .....                           | 397 |
| Configure Attributes at the Attribute Authority .....          | 398 |

---

|                                                                         |     |
|-------------------------------------------------------------------------|-----|
| Configure the Back Channel for the Attribute Authority .....            | 399 |
| Set up a SAML Requestor to Generate Attribute Queries .....             | 400 |
| Enable Attribute Queries and Specify Attributes .....                   | 401 |
| Configure the NameID for the Attribute Query .....                      | 402 |
| Configure the Backchannel for the Attribute Query .....                 | 402 |
| Create a Federation Attribute Variable .....                            | 403 |
| Create a Policy Expression with the Federation Attribute Variable ..... | 404 |

## **Chapter 16: Configure SiteMinder as an Account Partner 405**

|                                                                                          |     |
|------------------------------------------------------------------------------------------|-----|
| Configuration Checklist .....                                                            | 405 |
| Required Configuration Tasks for Configuring Resource Partners .....                     | 406 |
| Optional Configuration Tasks for Configuring a Resource Partner .....                    | 406 |
| Add a Resource Partner to an Affiliate Domain .....                                      | 407 |
| Select Users for Which Assertions Will Be Generated .....                                | 407 |
| Excluding a User or Group from Resource Partner Access .....                             | 408 |
| Allow Nested LDAP Groups Resource Partner Access .....                                   | 409 |
| Add Users by Manual Entry for Resource Partner Access .....                              | 409 |
| Specify Name IDs for WS-Federation Assertions .....                                      | 410 |
| Configure a Name ID for a WS-Federation Assertion .....                                  | 410 |
| Configure Required General Information for WS-Federation .....                           | 411 |
| Configure Single Sign-on for WS-Federation .....                                         | 412 |
| Assertion Validity for Single Sign-on .....                                              | 413 |
| Set the Authentication Scheme Protection Level .....                                     | 415 |
| Specify IP Address Restrictions for Resource Partners (optional) .....                   | 415 |
| Set up Time Restrictions for Resource Partner Availability (optional) .....              | 416 |
| Configure Attributes for WS-Federation Assertions (optional) .....                       | 417 |
| Configure Assertion Attributes for WS-Federation .....                                   | 417 |
| Allow Access to the Federation Web Services Application .....                            | 420 |
| Set Up Links to Initiate WS-Federation Single Sign-on .....                              | 421 |
| Configure Signout .....                                                                  | 422 |
| Enable Signout .....                                                                     | 422 |
| Validate Signout Requests that are Digitally Signed .....                                | 423 |
| Customizing Content in WS-Federation Assertions .....                                    | 424 |
| Integrate the Assertion Generator Plug-in with SiteMinder (SAML 2.0/WS-Federation) ..... | 425 |
| Protect the Authentication URL to Generate a SiteMinder Session .....                    | 427 |

## **Chapter 17: Configure SiteMinder as a Resource Partner 431**

|                                                            |     |
|------------------------------------------------------------|-----|
| WS-Federation Authentication Scheme Overview .....         | 431 |
| Configuration Tasks for WS-Federation Authentication ..... | 433 |
| WS-Federation Authentication Scheme Prerequisites .....    | 433 |
| Configure the WS-Federation Authentication Scheme .....    | 434 |

---

|                                                                                   |     |
|-----------------------------------------------------------------------------------|-----|
| Create a Custom WS-Federation Authentication Scheme .....                         | 435 |
| Locate User Records for Authentication .....                                      | 436 |
| Configure Disambiguation Locally .....                                            | 436 |
| Configure WS-Federation Single Sign-on at the Resource Partner .....              | 438 |
| Enforce a Single Use Policy to Enhance Security .....                             | 439 |
| Implement WS-Federation Signout .....                                             | 440 |
| Enable Signout .....                                                              | 441 |
| Customize Assertion Processing with the Message Consumer Plug-in .....            | 441 |
| Configure the Message Consumer Plug-in for WS-Federation .....                    | 442 |
| Integrate the Message Consumer Plug-in with SiteMinder (WS-Federation) .....      | 443 |
| Supply SAML Attributes as HTTP Headers .....                                      | 444 |
| Use Case for SAML Attributes As HTTP Headers .....                                | 444 |
| Configuration Overview to Supply Attributes as HTTP Headers .....                 | 446 |
| Set the Redirect Mode to Store SAML Attributes .....                              | 446 |
| Create an Authorization Rule to Validate Users .....                              | 447 |
| Configure a Response to Send Attributes as HTTP Headers .....                     | 448 |
| Create a Policy to Implement Attributes as HTTP Headers .....                     | 449 |
| Specify Redirect URLs for Failed WS-Federation Authentication .....               | 449 |
| How To Protect a Target Resource with a WS-Federation Authentication Scheme ..... | 450 |
| Assign the WS-Federation Scheme to a Realm .....                                  | 451 |
| Create a Rule for the Target WS-Federation Resource .....                         | 454 |
| Create a Policy for the WS-Federation Resource .....                              | 455 |

## **Chapter 18: Use SAML 2.0 Provider Metadata To Simplify Configuration 457**

|                                                                 |     |
|-----------------------------------------------------------------|-----|
| SiteMinder SAML 2.0 Metadata Tools Overview .....               | 457 |
| Export Metadata Tool .....                                      | 458 |
| Run the smfedexport Tool .....                                  | 461 |
| Command Options for smfedexport .....                           | 462 |
| smfedexport Tool Examples .....                                 | 464 |
| Import Metadata Tool .....                                      | 465 |
| Run the smfedimport Tool .....                                  | 466 |
| smfedimport Tool Examples .....                                 | 467 |
| Command Options for smfedimport .....                           | 468 |
| Processing Import Files with Multiple SAML 2.0 Providers .....  | 469 |
| Processing Import Files with Multiple Certificate Aliases ..... | 469 |

## **Chapter 19: Federation Security Services Trace Logging 471**

|                                                                 |     |
|-----------------------------------------------------------------|-----|
| Trace Logging .....                                             | 471 |
| Set Up and Enabling Trace Logging .....                         | 471 |
| Log Messages for Federation Web Services at the Web Agent ..... | 471 |
| Log Messages for Federation Services at the Policy Server ..... | 473 |

---

|                                                           |     |
|-----------------------------------------------------------|-----|
| Update Federation Web Services Data in the Logs .....     | 474 |
| Simplify Logging with Trace Configuration Templates ..... | 475 |
| Trace Logging Templates for Federation Web Services ..... | 475 |
| Trace Logging Templates for the IdP and SP .....          | 476 |

## **Chapter 20: Manage the Key Database for Signing and Encryption 479**

|                                                                                |     |
|--------------------------------------------------------------------------------|-----|
| SmKeyDatabase Overview .....                                                   | 479 |
| Role of the Smkeydatabase at the Producing Authority .....                     | 481 |
| Role of the Smkeydatabase at the Consuming Authority .....                     | 481 |
| Aliases in the Smkeydatabase .....                                             | 482 |
| Formats Supported by the Smkeydatabase .....                                   | 482 |
| What Gets Stored in smkeydatabase? .....                                       | 483 |
| Certificates Stored in the SmKeyDatabase Only at the Consuming Authority ..... | 484 |
| Certificate Revocation Lists in the smkeydatabase .....                        | 484 |
| Properties File for the Key Database .....                                     | 485 |
| Modify the Key Database Using smkeytool .....                                  | 486 |
| Smkeytool Command Syntax and Options .....                                     | 487 |
| Smkeytool Command Options .....                                                | 489 |
| Smkeytool Examples for UNIX Platforms .....                                    | 495 |
| Smkeytool Examples for Windows Platforms .....                                 | 497 |
| Migrate AM.keystore and Update smkeydatabase .....                             | 498 |
| Considerations Before Migrating Key Databases .....                            | 500 |
| Run the migratekeystore Tool .....                                             | 501 |

## **Chapter 21: Configuration Settings that Must Use the Same Values 503**

|                                                    |     |
|----------------------------------------------------|-----|
| How to Use the Configuration Settings Tables ..... | 503 |
| SAML 1.x Matching Configuration Settings .....     | 503 |
| SAML 2.0 Matching Configuration Settings .....     | 505 |
| WS-Federation Configuration Settings .....         | 506 |

## **Chapter 22: SAML 1.x Affiliate Dialog Reference 509**

|                                            |     |
|--------------------------------------------|-----|
| Affiliate Dialog .....                     | 509 |
| Affiliate Dialog Prerequisites .....       | 509 |
| Navigate to the Affiliates Dialog .....    | 509 |
| Affiliate Dialog Fields and Controls ..... | 510 |
| Affiliate Dialog--Users Tab .....          | 513 |
| Affiliate Dialog--Assertions Tab .....     | 514 |
| Affiliate Dialog--Session Tab .....        | 516 |
| Affiliate Dialog--Attributes Tab .....     | 516 |
| Affiliate Attribute Editor Dialog .....    | 517 |



---

|                                                              |     |
|--------------------------------------------------------------|-----|
| Affiliate Attribute Editor Prerequisites .....               | 517 |
| Navigate to the Affiliate Attribute Editor Dialog .....      | 517 |
| Affiliate Attribute Editor Dialog Fields and Controls .....  | 517 |
| Affiliate Attribute Editor Dialog--Attribute Setup Tab ..... | 518 |
| Affiliate Attribute Editor Dialog--Advanced Tab .....        | 519 |
| Affiliate Dialog--IP Addresses Tab .....                     | 520 |
| Affiliate Dialog--Time Tab .....                             | 520 |
| Affiliate Dialog--Advanced Tab .....                         | 521 |

## **Chapter 23: SAML 1.x Authentication Scheme Properties 523**

|                                                                                      |     |
|--------------------------------------------------------------------------------------|-----|
| Authentication Scheme Dialog Prerequisites .....                                     | 523 |
| Navigate to the Authentication Scheme Dialog .....                                   | 523 |
| Authentication Scheme Dialog--SAML Artifact Template .....                           | 524 |
| Authentication Scheme Dialog--SAML Artifact Template--Scheme Setup Tab .....         | 525 |
| Authentication Scheme Dialog--SAML Artifact Template--Advanced Tab .....             | 532 |
| Authentication Scheme Dialog--SAML POST Template .....                               | 532 |
| Authentication Scheme Dialog--SAML POST Template--Scheme Setup Tab .....             | 533 |
| Authentication Scheme Dialog--SAML POST Template--Advanced Tab .....                 | 538 |
| Authentication Scheme Dialog--SAML 1.x Auth Scheme Properties (Artifact, POST) ..... | 538 |

## **Chapter 24: SAML 2.0 Service Provider Reference 541**

|                                                                          |     |
|--------------------------------------------------------------------------|-----|
| SAML Service Provider Dialog .....                                       | 541 |
| Navigate to the SAML Service Provider Dialog .....                       | 541 |
| SAML Service Provider Dialog Prerequisites .....                         | 542 |
| SAML Service Provider Properties Dialog Fields and Controls .....        | 542 |
| SAML Service Provider Dialog--Users Tab .....                            | 545 |
| SAML Service Provider Dialog--Name IDs Tab .....                         | 545 |
| SAML Service Provider Dialog--General Tab .....                          | 547 |
| SAML Service Provider Dialog--Backchannel Properties .....               | 549 |
| SAML Service Provider Dialog--SSO Tab .....                              | 550 |
| Assertion Consumer Services Dialog .....                                 | 553 |
| Restrictions Dialog for an SP .....                                      | 553 |
| Navigate to the Restrictions Dialog .....                                | 553 |
| Restrictions Dialog Prerequisites .....                                  | 554 |
| Restrictions Dialog Fields and Controls .....                            | 554 |
| SAML Service Provider Dialog--Attributes Tab .....                       | 555 |
| Navigating to the SAML Service Provider Attribute Editor Dialog .....    | 555 |
| SAML Service Provider Attribute Editor Dialog Prerequisites .....        | 556 |
| SAML Service Provider Attribute Editor Dialog--Attribute Setup Tab ..... | 556 |
| SAML Service Provider Attribute Editor Dialog--Advanced Tab .....        | 558 |
| SAML Service Provider Dialog--SLO Tab .....                              | 558 |

---

|                                                     |     |
|-----------------------------------------------------|-----|
| SAML Service Provider Dialog--IPD Tab .....         | 559 |
| SAML Service Provider Dialog--Encryption Tab .....  | 560 |
| SAML Service Provider Dialog--Advanced Tab .....    | 561 |
| SAML 2.0--Additional URL Configuration Dialog ..... | 562 |

## **Chapter 25: SAML 2.0 Authentication Scheme Properties Reference 565**

|                                                                  |     |
|------------------------------------------------------------------|-----|
| Authentication Scheme Dialog -- SAML 2.0 Template .....          | 565 |
| Navigate to the SAML 2.0 Auth Scheme Properties Dialog .....     | 566 |
| SAML 2.0 Authentication Properties--Scheme Setup Tab .....       | 566 |
| SAML 2.0 Authentication Properties--Advanced Tab .....           | 568 |
| SAML 2.0 Auth Scheme Properties Dialog .....                     | 568 |
| SAML 2.0 Auth Scheme Properties Dialog Prerequisites .....       | 568 |
| SAML 2.0 Auth Scheme Properties Dialog Fields and Controls ..... | 568 |
| SAML 2.0 Auth Scheme Properties Dialog--Users Tab .....          | 569 |
| SAML 2.0 Auth Scheme Properties Dialog--SSO Tab .....            | 571 |
| SAML 2.0 Auth Scheme Properties Dialog--SLO Tab .....            | 575 |
| SAML 2.0 Auth Scheme Properties Dialog--Encryption Tab .....     | 577 |
| SAML 2.0 Auth Scheme Properties Dialog--Attributes Tab .....     | 577 |
| SAML 2.0 Auth Scheme Properties Dialog--Name IDs Tab .....       | 577 |
| SAML 2.0 Auth Scheme Properties Dialog--Backchannel Tab .....    | 578 |
| SAML 2.0 Auth. Scheme Properties Dialog--Advanced Tab .....      | 579 |
| SAML 2.0--Additional URL Configuration .....                     | 581 |

## **Chapter 26: SAML Affiliations Reference 583**

|                                                   |     |
|---------------------------------------------------|-----|
| SAML Affiliation Properties Dialog .....          | 583 |
| SAML Affiliation Dialog Prerequisites .....       | 583 |
| Navigate to the SAML Affiliation Dialog .....     | 583 |
| SAML Affiliation Dialog Fields and Controls ..... | 584 |
| SAML Affiliation Dialog--Name IDs Tab .....       | 585 |
| SAML Affiliation Dialog--Users Tab .....          | 586 |
| SAML Service Providers Tab .....                  | 588 |
| SAML Auth Schemes Tab .....                       | 588 |

## **Chapter 27: Attribute Authority and Query Reference 589**

|                                                |     |
|------------------------------------------------|-----|
| SAML 2.0 Attribute Authority Reference .....   | 589 |
| Attribute Svc Dialog .....                     | 589 |
| Attribute Svc Dialog Prerequisites .....       | 589 |
| Navigate to the Attribute Svc Dialog .....     | 589 |
| Attribute Svc Dialog Fields and Controls ..... | 590 |
| SAML 2.0 Attribute Query Reference .....       | 591 |

---

|                                                      |     |
|------------------------------------------------------|-----|
| Attributes Dialog Prerequisites .....                | 591 |
| Navigate to the Attributes Dialog .....              | 592 |
| Attributes Dialog for Attribute Queries .....        | 592 |
| Name IDs Dialog Fields for Attribute Queries .....   | 594 |
| Federation Attribute Variable Editor Reference ..... | 596 |

## **Chapter 28: WS-Federation Resource Partner Properties Reference 597**

|                                                              |     |
|--------------------------------------------------------------|-----|
| Resource Partner Properties Dialog .....                     | 597 |
| Resource Partner Properties Dialog Prerequisites .....       | 597 |
| Navigate to the Resource Partner Properties Dialog .....     | 598 |
| Resource Partner Properties Dialog Fields and Controls ..... | 598 |
| Resource Partner Properties Dialog--Users Tab .....          | 601 |
| Resource Partner Properties Dialog--Name IDs Tab .....       | 602 |
| Resource Partner Properties Dialog--General Tab .....        | 603 |
| Resource Partner Properties Dialog--SSO Tab .....            | 604 |
| Restrictions Dialog for a Resource Partner .....             | 606 |
| Restrictions Dialog Prerequisites .....                      | 606 |
| Restrictions Dialog Field and Controls .....                 | 606 |
| Resource Partner Properties Dialog--Attributes Tab .....     | 607 |
| Resource Partner Attribute Dialog Prerequisites .....        | 607 |
| Resource Partner Attribute Dialog--Attribute Setup Tab ..... | 607 |
| Resource Partner Attribute Dialog--Advanced Tab .....        | 609 |
| Resource Partner Properties Dialog--Signout Tab .....        | 609 |
| Resource Partner Properties Dialog--Advanced Tab .....       | 610 |

## **Chapter 29: WS-Federation Authentication Scheme Reference 611**

|                                                                              |     |
|------------------------------------------------------------------------------|-----|
| Authentication Scheme Properties Dialog--WS-Federation Template .....        | 611 |
| WS-Federation Auth. Scheme Properties Dialog Prerequisites .....             | 612 |
| Navigate to the Auth. Scheme Properties Dialog--WS Federation Template ..... | 612 |
| Federation Authentication Scheme Dialog--Scheme Setup Tab .....              | 613 |
| WS-Federation Authentication Scheme Dialog--Advanced Tab .....               | 614 |
| WS-Federation Auth. Scheme Fields and Controls .....                         | 615 |
| Auth. Scheme Properties--WS-Federation--Users Tab .....                      | 616 |
| Auth. Scheme Properties--WS-Federation--SSO Tab .....                        | 617 |
| Auth. Scheme Properties--WS-Federation--Signout Tab .....                    | 619 |
| Auth. Scheme Properties--WS-Federation--Advanced Tab .....                   | 620 |

## **Chapter 30: Federation Web Services URLs Used in SiteMinder Configuration 623**

|                                |     |
|--------------------------------|-----|
| Federation Services URLs ..... | 623 |
|--------------------------------|-----|

---

|                                                              |     |
|--------------------------------------------------------------|-----|
| URLs for Services at the Producing Authority .....           | 623 |
| Intersite Transfer Service (SAML 1.x) .....                  | 624 |
| Assertion Retrieval Service (SAML 1.x) .....                 | 625 |
| Artifact Resolution Service (SAML 2.0) .....                 | 626 |
| Single Sign On Service (SAML 2.0) .....                      | 627 |
| Single Logout Service at the IdP (SAML 2.0) .....            | 628 |
| Identity Provider Discovery Profile Service (SAML 2.0) ..... | 629 |
| Attribute Service (SAML 2.0) .....                           | 629 |
| Single Sign-on Service (WS-Federation) .....                 | 631 |
| Signout Service at the AP (WS-Federation) .....              | 632 |
| WSFedDispatcher Service at the AP .....                      | 633 |
| URLs for Services at the Consuming Authority .....           | 633 |
| SAML Credential Collector (SAML 1.x) .....                   | 634 |
| AuthnRequest (SAML 2.0) .....                                | 635 |
| Assertion Consumer Service (SAML 2.0) .....                  | 636 |
| Security Token Consumer Service (WS-Federation) .....        | 637 |
| Single Logout Service at the SP (SAML 2.0) .....             | 638 |
| Signout Service at the RP (WS-Federation) .....              | 639 |
| WSFedDispatcher Service at the RP .....                      | 640 |
| The Web.xml File .....                                       | 640 |

## Chapter 31: Troubleshooting 641

|                                                                                |     |
|--------------------------------------------------------------------------------|-----|
| General Issues .....                                                           | 641 |
| Web Agent Option Pack Fails to Initialize Due to Invalid smjavaagent.dll ..... | 641 |
| Cookie Domain Mismatch Errors .....                                            | 641 |
| Error After Successful Authentication at Consumer/SP .....                     | 642 |
| HTTP 404 Error When Trying to Retrieve Assertion at the Consumer .....         | 642 |
| Federation Web Services Fails to Send SAML Request to Producer/IdP .....       | 642 |
| Matching Parameter Case-Sensitivity Configuration Issues .....                 | 643 |
| Error Message When Viewing FederationWSCustomUserStore .....                   | 643 |
| Policy Server System Fails After Logout .....                                  | 643 |
| Encrypted Private Key Fails to Be Imported into SMkeydatabase .....            | 644 |
| Multibyte Characters in Assertions are Not Handled Properly .....              | 644 |
| Trace Logs Not Appearing for IIS Web Server Using ServletExec .....            | 644 |
| Error During Initialization of JVM .....                                       | 645 |
| Affwebserver.log and FWSTrace.log Show Wrong Time .....                        | 645 |
| SAML 1.x-Only Issues .....                                                     | 646 |
| SAML 1.x Artifact Profile Single Sign-On Failing .....                         | 646 |
| Consumer Not Authenticating When Accessing Assertion Retrieval Service .....   | 646 |
| Authentication Fails After Modifying Authentication Method .....               | 647 |
| Client Authentication Fails for SAML Artifact Single Sign-on .....             | 647 |

---

|                                                                        |     |
|------------------------------------------------------------------------|-----|
| SAML 2.0-Only Issues .....                                             | 647 |
| SP Not Authenticating When Accessing Assertion Retrieval Service ..... | 648 |
| ODBC Errors Deleting Expiry Data From Session Server .....             | 648 |

|              |            |
|--------------|------------|
| <b>Index</b> | <b>649</b> |
|--------------|------------|



# Chapter 1: Federation Security Services Overview

---

This section contains the following topics:

- [Introduction to SiteMinder Federation Security Services](#) (see page 23)
- [Federation Use Cases](#) (see page 24)
- [Solutions for Federation Use Cases](#) (see page 36)
- [Federation Security Services Concepts](#) (see page 71)
- [SAML Profiles Supported by SiteMinder](#) (see page 75)
- [Benefits of SiteMinder Federation Security Services](#) (see page 75)
- [SiteMinder Components for Federation Security Services](#) (see page 77)
- [Federation Security Services Process Flow](#) (see page 85)
- [Where to Find Federation Security Services Information](#) (see page 111)

## 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.

## Federation Use Cases

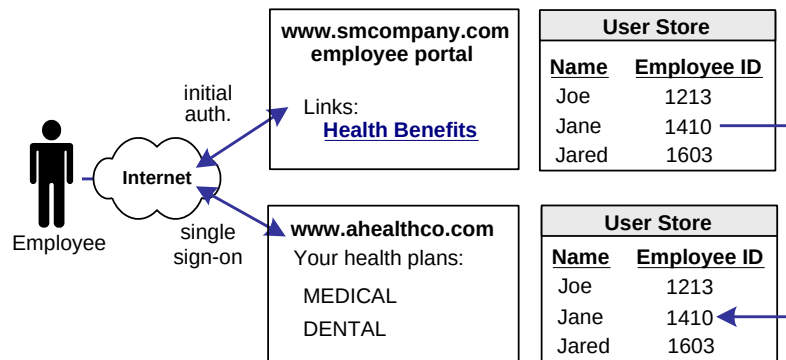
There are probably as many use cases for federated networks as there are business arrangements between partners. This section presents use cases that demonstrate different ways of handling user identities to provide single sign-on and single logout between partners.

### 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 ahealthco.com's web site and is presented with her health benefit information without having to sign on to ahealthco.com's Web site.

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.

#### More information:

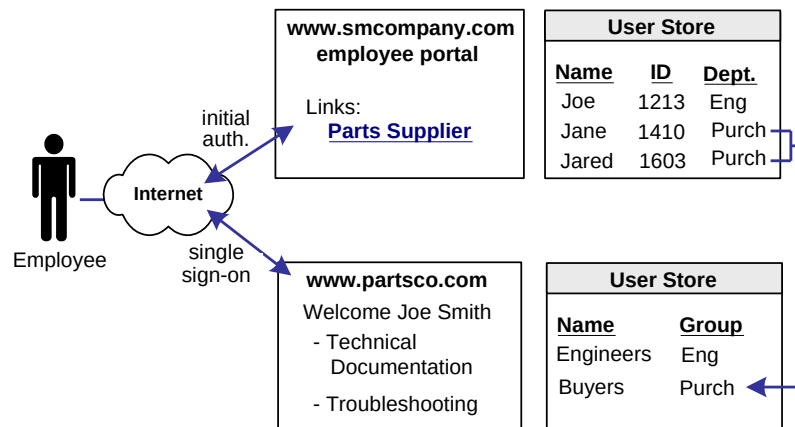
[Solution 1: Single Sign-on based on Account Linking](#) (see page 37)



## 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 his employee portal, smcompany.com and clicks a link to access information at partsco.com. Because the user is an engineer at smcompany.com, he is taken directly to the Specifications and Parts List portion of partsco.com's web site without having to sign in.



When a buyer for smcompany.com authenticates at smcompany.com and clicks a link to access information at partsco.com, she is taken directly to the ordering area of partsco.com's web site without having to sign on.

Additional attributes, such as 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 access to sensitive portions of the Partsco.com Web site must be controlled. To do this, 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, user attributes are sent in a secure manner from smcompany.com to partsco.com, which uses them to determine what profile identity should be used to control access.

### More information:

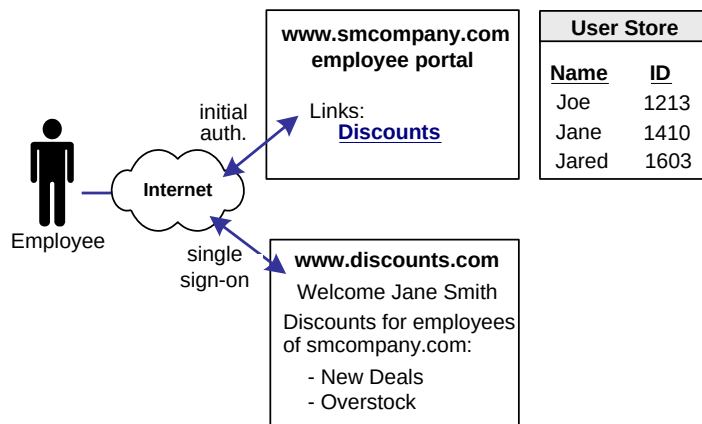
[Solution 2: Single Sign-on based on User Attribute Profiles](#) (see page 44)

## 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 an employee portal at www.smcompany.com and clicks a link to access discounts at discounts.com. The employee is taken to discounts.com's web site and presented with the discounts available for smcompany.com employees, without having to sign on to discounts.com's Web site.

The following illustration shows this use case.



Discounts.com does not maintain any identities for employees of smcompany.com—the company allows all employees of smcompany.com to access discounts.com 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.

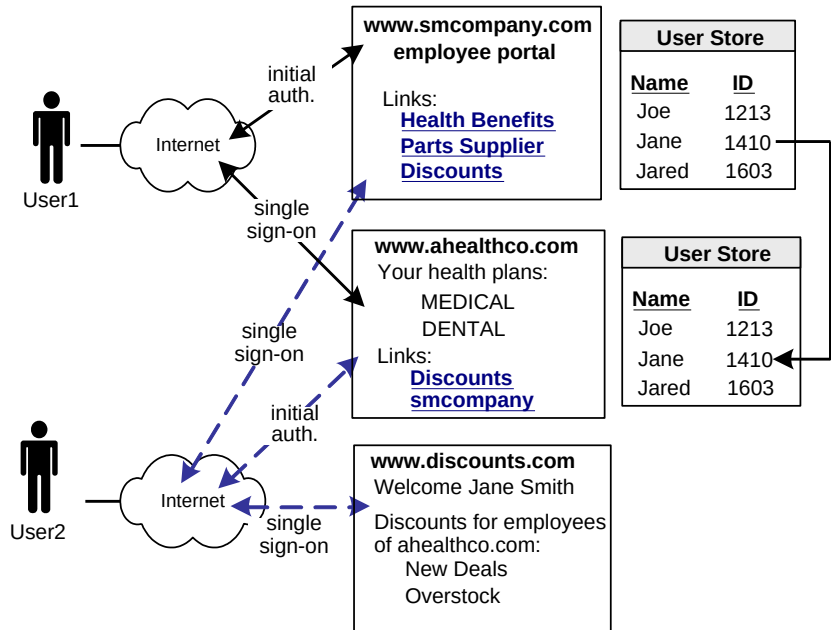
Additional attributes, such as user name are passed from smcompany.com to discounts.com to personalize the interface for the individual user.

### More information:

[Solution 3: Single Sign-on with no Local User Account](#) (see page 46)

## 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 use cases presented previously.



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

In this extended network, the users access each Web site differently:

- User1 accesses health benefit information at ahealthco.com's web site. User 1 may also choose to access partsco.com's Web site by clicking on the PartsSupplier link at smcompany.com, her employee portal. She can also click a link at her employee portal to access discounts at discounts.com.

- User2 authenticates at ahealthco.com's web site and clicks a link to access discounts at discounts.com, without having to sign on to discounts.com's web site. The discounts presented to User2 reflect the business arrangement between ahealthco.com and discounts.com. Because User2 is an employee of smcompany.com, he can also click a link at ahealthco.com and access the employee portal at smcompany.com without having to sign on to smcompany.com's web site.
- User3 (not shown in example), is a customer of ahealthco.com, but is not an employee of smcompany.com. User3 authenticates at ahealthco.com's web site and clicks a link to access discounts at discounts.com without having to sign on to discounts.com's web site. The discounts presented to User3 reflect the business arrangement between ahealthco.com and discounts.com. Since User3 is not an employee of smcompany.com, User3 cannot access smcompany.com's web site.

**More information:**

[Solution 4: Extended Networks](#) (see page 49)

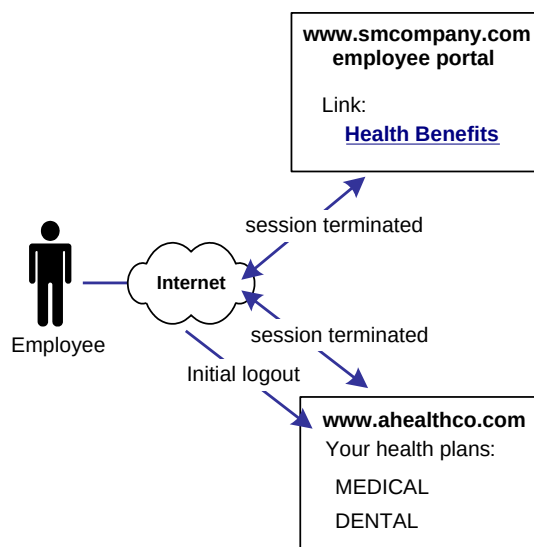
## Use Case 5: Single Logout

In Use Case 5, an employee of smcompany.com authenticates at an employee portal, www.smcompany.com, and selects a link to view her health benefits at www.ahhealthco.com. The employee is taken to ahealthco.com's Web site and presented with her health benefit information without having to sign on to ahealthco.com's Web site.

After the employee has finished looking at her health benefits, ahealthco.com wants to ensure that when the employee logs out from ahealthco.com, the user's session at ahealthco.com and the session at smcompany.com is terminated. Terminating both sessions ensures that an unauthorized employee cannot use the existing sessions to access resources at smcompany.com or to view benefits of the authorized employee.

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

The following illustration shows the use case.



#### More information

[Solution 5: Single Logout \(SAML 2.0\)](#) (see page 50)

## Use Case 6: WS-Federation Signout

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

After the employee has finished looking at her health benefits, ahealthco.com wants to ensure that when the employee logs out from ahealthco.com, the user's session at ahealthco.com and the session at smcompany.com is terminated. Terminating both sessions ensures that an unauthorized employee cannot use the existing sessions to access resources at smcompany.com or to view benefits of the authorized employee.

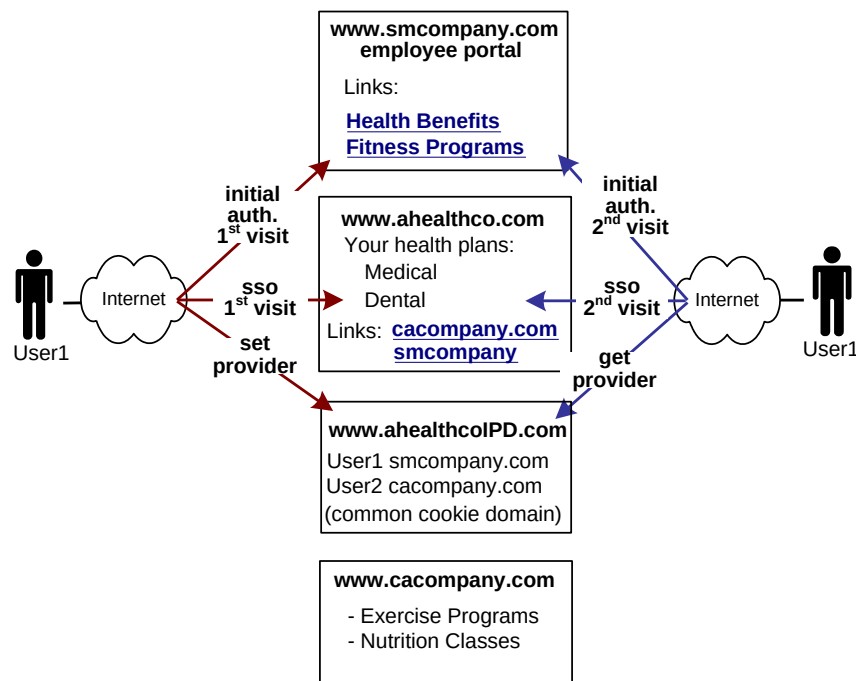
#### More Information

[Solution 6: WS-Federation Signout](#) (see page 52)

## Use Case 7: Identity Provider Discovery Profile

In Use Case 7, several companies, such as smcompany.com contract health benefits from ahealthco.com. Ahealthco.com wants to determine which company users are coming from so it can send the user back to the correct company to log on.

The following illustration shows a network where Identity Provider Discovery Profile is used.



When a user arrives at ahealthco.com, this health provider wants to determine which site to send the user so the user can log on. For User1, smcompany.com is the company where this user logs on, smcompany.com is set in the common domain cookie. For another user, cacompany.com is another Identity Provider at which a user can authenticate and then cacompany.com will be set in the common domain cookie at ahealthco.com.

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

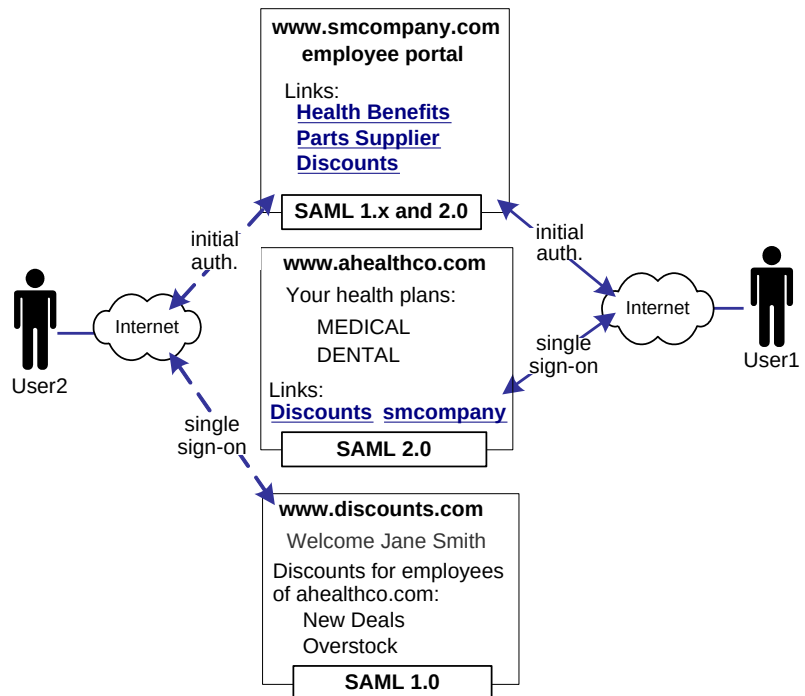
### More Information:

[Solution 7: Identity Provider Discovery Profile \(SAML 2.0\)](#) (see page 54)

## 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 particular protocol used by the SP consuming the assertion.

The following illustration shows the multiprotocol use case.



### More Information:

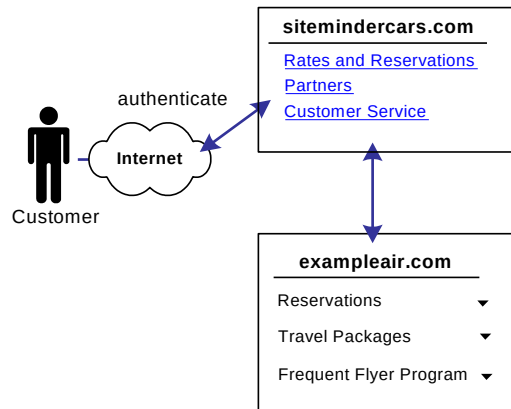
[Solution 8: Multi-protocol Network](#) (see page 56)

## 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 www.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 customer's frequent flyer number with exampleair.com. The customer's frequent flyer 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 based on the customer's frequent flyer number instead of requiring the customer to sign-on and authenticate at exampleair.com.

**More Information:**

[Solution 9: SAML 2.0 User Authorization Based on a User Attribute](#) (see page 58)

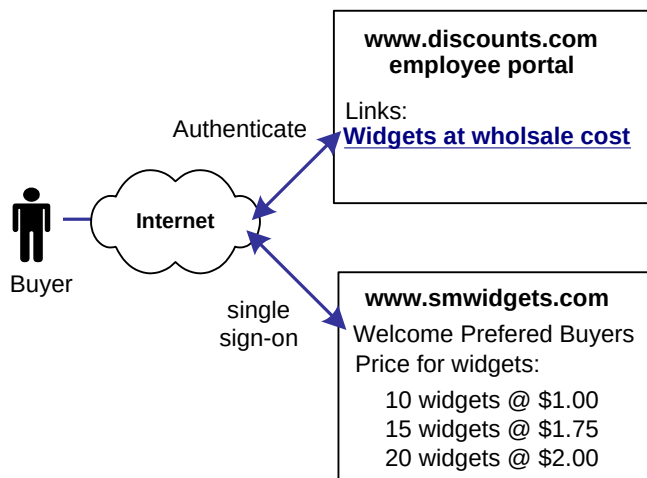
## Use Case 10: SAML 2.0 Single Sign-on with No Name ID at the IdP

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

A buyer for discounts.com clicks on a link to access the latest price list on widgets at smwidgets.com. The buyer is taken to smwidgets.com's Web site and presented with the price list without having to sign on to discounts.com's Web site.



The following illustration shows this use case.



Discounts.com requests access to a price list at smwidgets.com. There is no federated user identity stored at discounts.com for its buyers or at smwidgets.com. When a request from discounts.com is sent to smwidgets.com, this entity creates an identity for the buyer that it sends back in a secure manner to discounts.com. Discounts.com uses this identity to authenticate the user and allow the buyer access to the requested resource.

#### More Information:

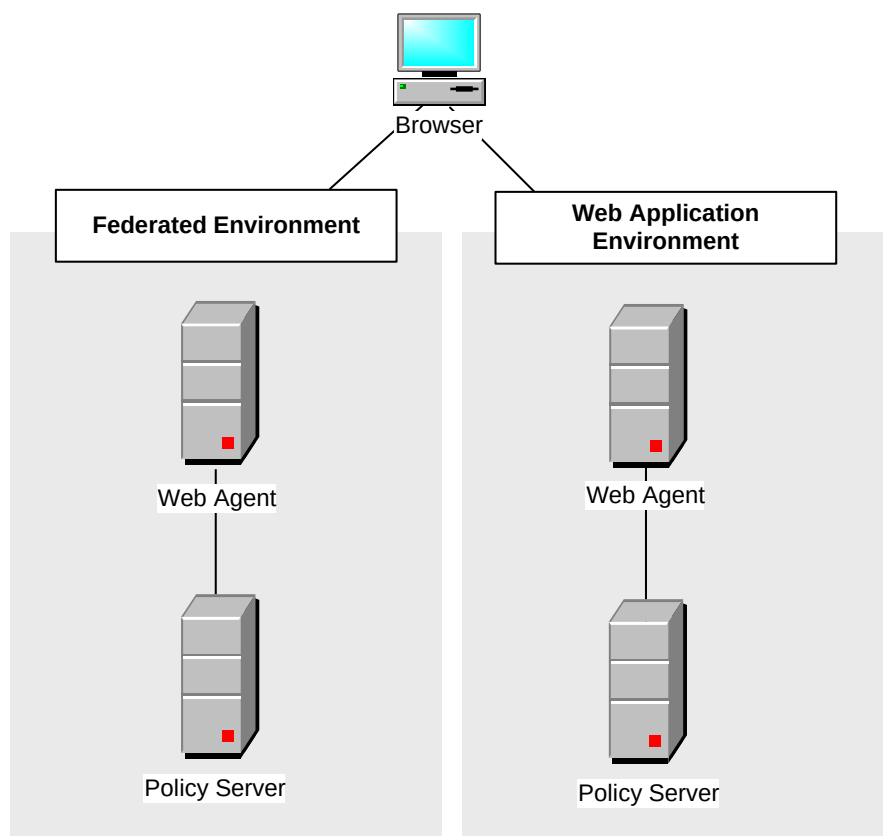
[Solution 10: Single Sign-on with No User ID at the IdP](#) (see page 59)

## 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 the 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 must be made to the session store to retrieve the assertion, impacting performance.

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



**More Information:**

[Solution 11: SAML Artifact SSO Using Security Zones](#) (see page 61)

## 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 attributes in an assertion that are from a Web application. 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 protocols IdPA.com uses is SAML 2.0 (POST and Artifact) and WS-Federation.

The following figure shows an example of attributes gathered from a Web application and used for single sign-on.



#### IdP-initiated Single Sign-on with Web Application Attributes

IdPA.com has created a Web application for access to protected resources at its business partner SPB.com. When the customer logs in at IdPA.com, they click on a link for the business partner and they are sent to the Web application, where they are prompted to enter a customer ID. This information needs to be sent to SPB.com so that the customer is permitted access to the appropriate services.

#### SP-initiated Single Sign-on with Web Application Attributes

A customer has clicked on a link at SPB.com, the Service Provider. This link is for protected resources, so he is redirected to IdpA.com to be authenticated. After the user successfully authenticates at IdPA.com, he is redirected to the Web application where he provides specific user information. Upon submitting the information, the customer is sent back to SPB.com to complete single sign-on for the requested resource.

#### More information:

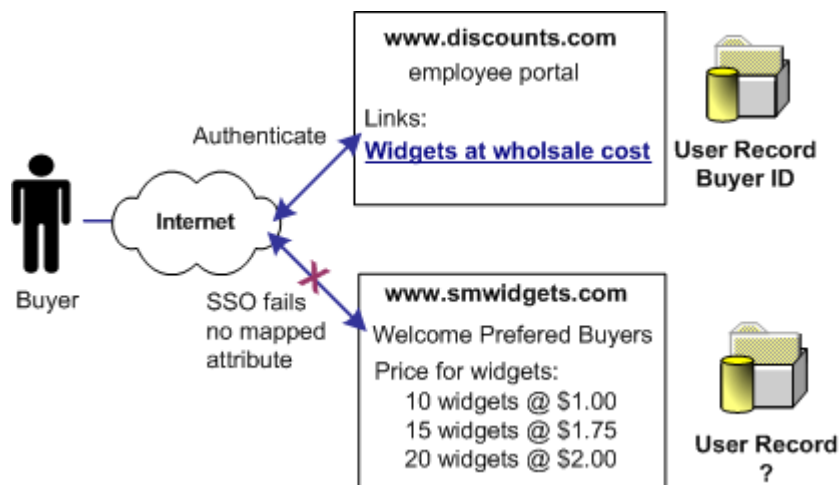
[Solution 12: SAML 2.0 SSO with Attributes from a Web Application](#) (see page 64)

## 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 his name and buyer ID.

The following illustration shows this use case.



The identity based on the user's buyer ID 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 has to establish a mapping using dynamic account linking so that smwidgets can authenticate the employee and allow access to the price list.

### More information:

[Solution 13: SAML 2.0 SSO with Dynamic Account Linking at the SP](#) (see page 68)

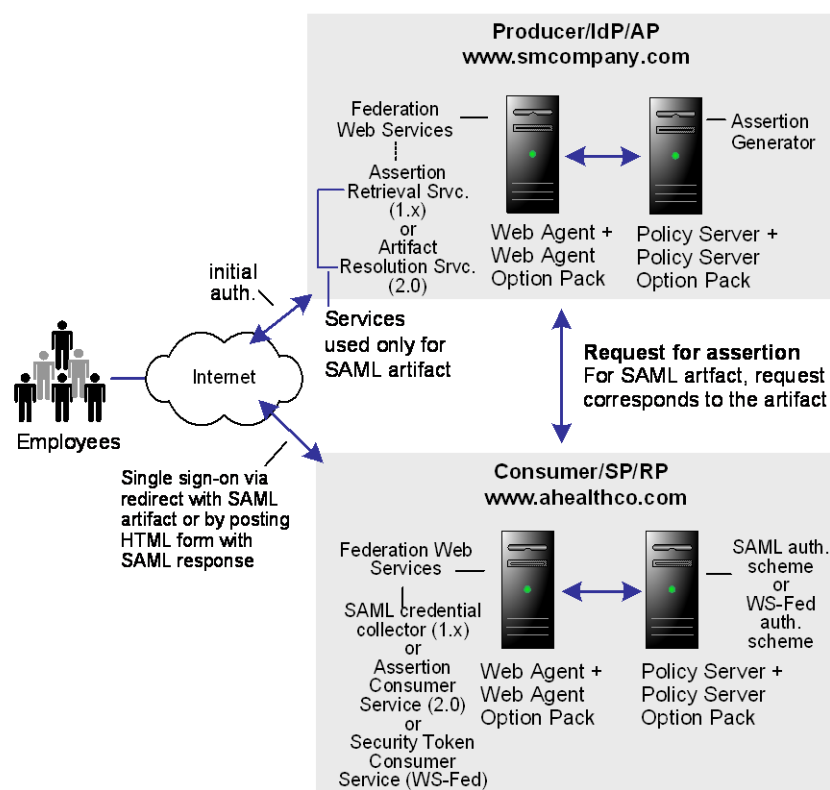
## Solutions for Federation Use Cases

SiteMinder Federation Security Services components work together to take advantage of the growth of e-business networks and offer enhanced services to employees, customers and suppliers.

SiteMinder offers solutions on how these components solve the use cases in this guide.

## 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 24).



SiteMinder v6.x is deployed at both sites. The Web Agent with the Web Agent Option Pack are installed on a Web server machine and the Policy Server with the Policy Server Option Pack are installed on another machine. The installations are the same for both smcompany.com and ahealthco.com.

**Note:** The SPS federation gateway can replace the Web Agent and Web Agent Option Pack to provide the SiteMinder Federation Web Services application functions. For information about installing and configuring the SPS federation gateway, see the *CA SiteMinder 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 her 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, inserts the assertion into the SiteMinder session server, and returns a SAML artifact.
4. The Web Agent redirects the user to www.ahhealthco.com with the SAML artifact, in accordance with the SAML browser artifact protocol.

Ahealthco.com is acting as the consumer site. The redirect request with the SAML artifact is handled by the SAML credential collector service that is part of the Federation Web Services 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 server 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 and proceeds to issue a SiteMinder session cookie to the user's browser.
5. At this point the user is allowed access to resources at ahealthco.com based on policies defined at the Policy Server at ahealthco.com and enforced by the Web Agent at ahealthco.com.

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

The administrator at ahealthco.com uses the Policy Server User Interface to configure a SAML artifact authentication scheme for smcompany.com. The authentication scheme specifies the location of the assertion retriever service at smcompany.com, how to extract the unique user ID from the SAML assertion, and how to search the user directory at ahealthco.com for the user record that matches the value extracted 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 her 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 user's browser.
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 redirect request with the SAML response is handled by the SAML credential collector service 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, which is protected by the SAML POST profile authentication scheme.
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 logging in, the SAML credential collector creates an SMSESSION cookie, places it in the user's browser, and redirects the user to the target resource at ahealthco.com.
6. At this point the user is allowed access to resources at ahealthco.com based on policies defined at the Policy Server and enforced by the Web Agent at ahealthco.com.

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

The administrator at ahealthco.com uses the Policy Server User Interface to configure a SAML POST profile authentication scheme for smcompany.com. The authentication scheme specifies how to extract the unique user ID from the SAML assertion, and how to search the user directory at ahealthco.com for the user record that matches the value extracted 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 is referred to as an unsolicited response at the Identity Provider.
2. When the employee clicks a link at www.smcompany.com to view her 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, which creates a SAML assertion, inserts the assertion into the SiteMinder session server, 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. The redirect request with the SAML artifact is handled by the Assertion Consumer Service that is part of the Federation Web Services at ahealthco.com.

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 SiteMinder session server at smcompany.com and returns it to the artifact resolution service at ahealthco.com.
4. The Assertion Consumer Service then passes the assertion to the SAML 2.0 authentication scheme for validation and session creation and proceeds to issue a SiteMinder session cookie to the user's browser.
5. At this point, the user is allowed access to resources at ahealthco.com based on policies defined at the Policy Server at ahealthco.com and enforced by the Web Agent at ahealthco.com.

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

The administrator at ahealthco.com uses the Policy Server User Interface to configure a SAML 2.0 authentication scheme that uses the artifact binding for smcompany.com. The authentication scheme specifies the location of the artifact resolution service at smcompany.com, how to extract the unique user ID from the SAML assertion, and how to search the user directory at ahealthco.com for the user record that matches the value extracted from 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 is referred to as an unsolicited response at the Identity Provider.
2. When the employee clicks a link at www.smcompany.com to view her 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 user's browser.
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 redirect request with the SAML response is handled by the Assertion Consumer Service, which 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. This resource is protected by the SAML 2.0 authentication scheme 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 logging in, the Assertion Consumer Service creates an SMSESSION cookie, places it in the user's browser, and redirects the user to the target resource at ahealthco.com.
5. At this point the user is allowed access to resources at ahealthco.com based on policies defined at the Policy Server and enforced by the Web Agent at ahealthco.com.

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

The administrator at ahealthco.com uses the Policy Server User Interface to configure a SAML 2.0 authentication scheme with the HTTP-POST binding for smcompany.com. The authentication scheme specifies how to extract the unique user ID from the SAML assertion, and how to search the user directory at ahealthco.com for the user record that matches the value extracted 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. It 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 user's browser.
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 redirect request with the SAML response is handled by the Security Token Consumer Service, which 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. This resource is protected by the WS-Federation authentication scheme.
2. Because the WS-Federation authentication scheme is protecting the resource, the Security Token Consumer Service passes the signed assertion in the 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 logging in, the Security Token Consumer Service creates an SMSESSION cookie, places it in the user's browser, and redirects the user to the target resource at ahealthco.com.
5. At this point the user is allowed access to resources at ahealthco.com based on policies defined at the Policy Server and enforced by the Web Agent at ahealthco.com.

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

The administrator at ahealthco.com uses the Policy Server User Interface to configure a WS-Federation authentication scheme for smcompany.com. The authentication scheme specifies how to extract the unique user ID from the SAML assertion, and how to search the user directory at ahealthco.com for the user record that matches the value extracted from the assertion.

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

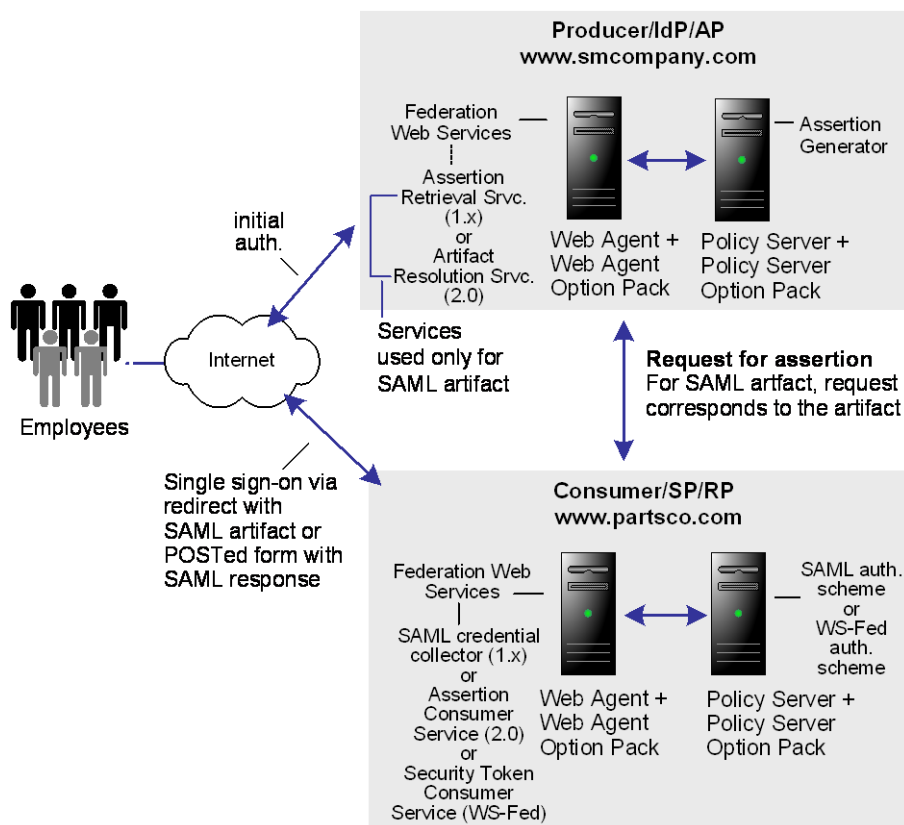
Solution 2 shows how SiteMinder Federation Security Services can be deployed at smcompany.com and partsco.com to solve [Use Case 2: Single Sign-on Based on User Attribute Profiles](#) (see page 25).

SiteMinder v6.x is deployed at both sites. The interactions between the user and each site is similar, where partsco.com is acting as the consuming authority.

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 (for artifact profile only) is at the IdP and the SAML credential collector is at the SP.
- For SAML 2.0, the Assertion Retrieval Service (for 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 SiteMinder Federation Web Services application functions. For information about installing and configuring the SPS federation gateway, see the *CA SiteMinder Secure Proxy Server Administration Guide*.

The configuration is similar to [Solution 1: Single Sign-on based on Account Linking](#) (see page 37), except for the following:

- The administrator at smcompany.com defines the consumer/SP for partsco.com with an attribute specifying the user's department at the company. The assertion generator will include this attribute as part of the user profile in the SAML assertion created 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 and searches the user directory at partsco.com for the user record that matches the department value taken from the assertion. The administrator defines one user profile record for each department that is allowed to access partsco.com's Web site.

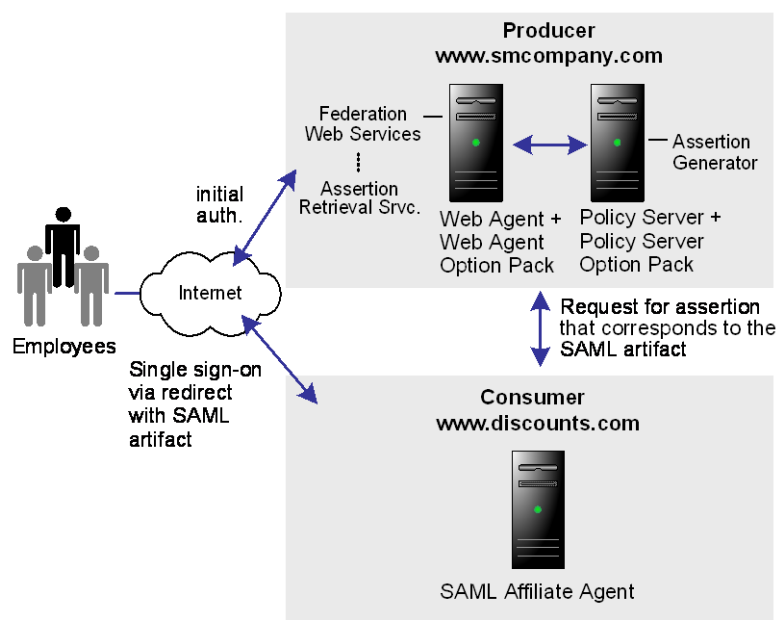
## Solution 3: Single Sign-on with no Local User Account

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 26).

SiteMinder v6.x is deployed at smcompany.com by installing the Web Agent with the Web Agent Option pack on one machine, and installing the Policy Server with the Policy Server Option Pack on another machine. The SAML Affiliate Agent is installed at discounts.com.

**Note:** The SAML Affiliate Agent only supports SAML 1.0.

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 SiteMinder Federation Web Services application functions. For information about installing and configuring the SPS federation gateway, see the *CA SiteMinder 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 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, which creates a SAML assertion, inserts the assertion into the SiteMinder session server, and returns a SAML artifact.
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 redirect request with the SAML artifact is handled by the SAML Affiliate Agent at [www.discounts.com](http://www.discounts.com), as follows:

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

The administrator at [smcompany.com](http://smcompany.com) uses the Policy Server User Interface to configure an affiliate for [discounts.com](http://discounts.com). The affiliate is configured with attribute information to be passed to [discounts.com](http://discounts.com). The assertion generator will include those attributes as part of the user profile in a SAML assertion created for [discounts.com](http://discounts.com).

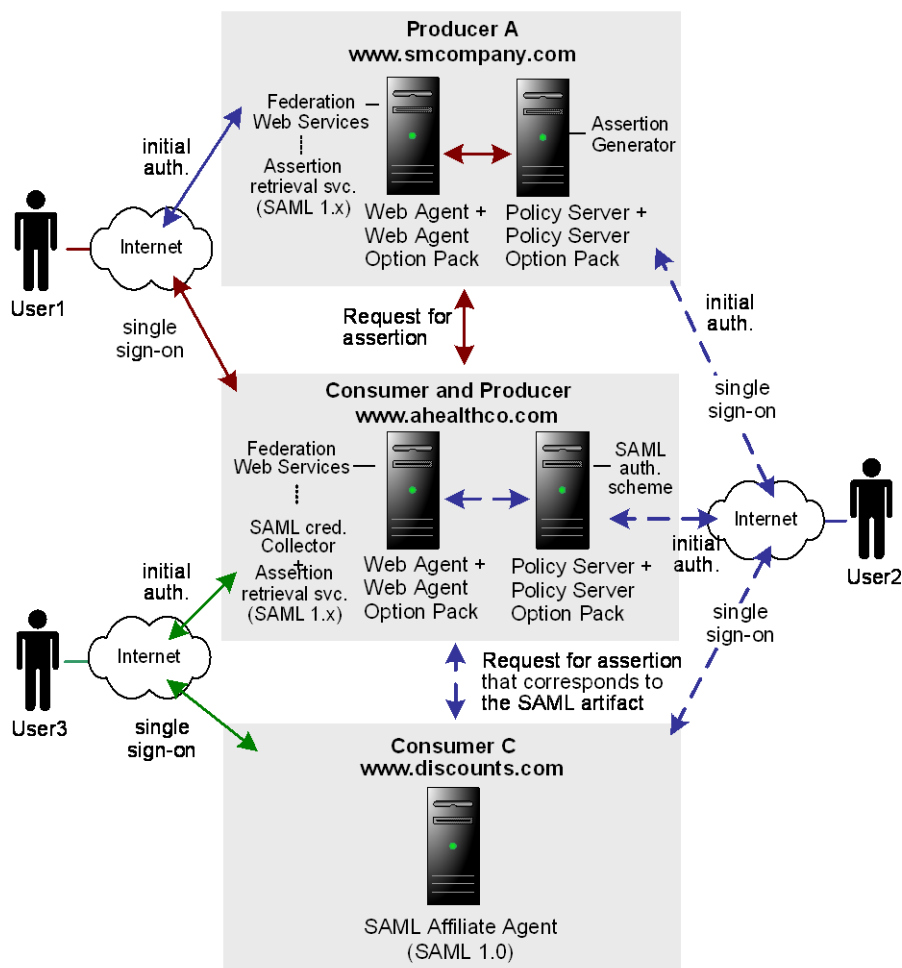
The administrator at [discounts.com](http://discounts.com) configures the SAML Affiliate Agent with information about the [discounts.com](http://discounts.com) site, the location of the assertion retriever service at [smcompany.com](http://smcompany.com), and what resources are to be protected by the affiliate defined at [smcompany.com](http://smcompany.com).



## Solution 4: Extended Networks

Solution 4 illustrates how SiteMinder Federation Security Services can be deployed at [smcompany.com](http://smcompany.com), [ahealthco.com](http://ahealthco.com), and [discounts.com](http://discounts.com) to solve [Use Case 4: Extended Networks](#) (see page 27).

The following illustration shows an extended network. SAML 1.x is the protocol being used.



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

SiteMinder is deployed at smcompany.com and ahealthco.com by installing the Web Agent with the Web Agent Option Pack on one machine, and the Policy Server with the Policy Server Option Pack on another machine. 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 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 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.

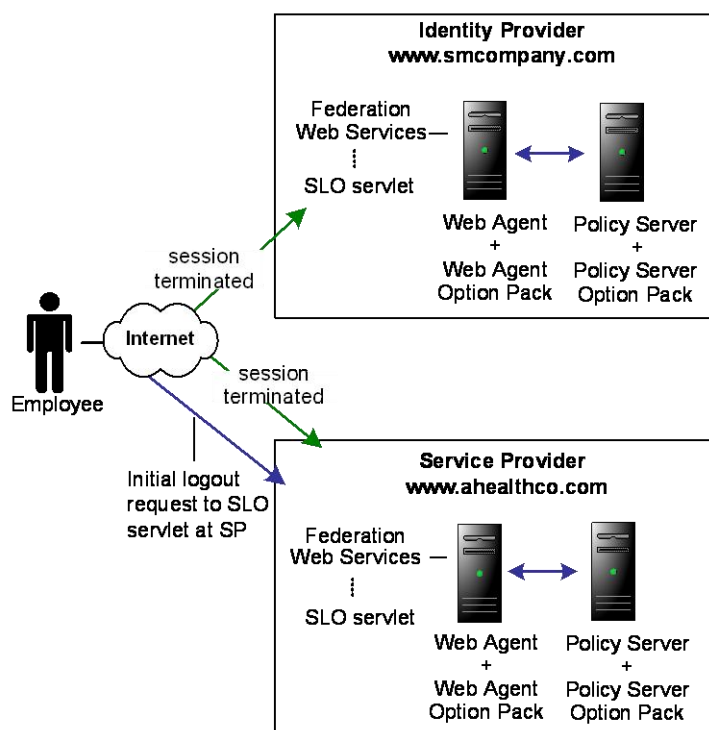
## Solution 5: Single Logout (SAML 2.0)

Solution 5 illustrates how SiteMinder Federation Security Services can be employed to solve [Use Case 5: Single Logout](#) (see page 28).

In this solution:

- smcompany.com is the Identity Provider
- ahealthco is the Service Provider that initiates the logout request.
- Single logout is enabled using the Policy Server User Interface 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 SiteMinder Federation Web Services application functions. For 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. Employee performs single sign-on between smcompany.com and ahealthco.com. Smcompany.com places information about ahealthco.com in its session server. Ahealthco.com places information about smcompany.com in its session server.
2. After the employee has finished looking at her health benefits, she clicks a log out link at the Service Provider. This user's browser accesses the single logout servlet at the Service Provider.
3. The user's session is terminated from the Service Provider's session store.  
**Note:** This does not remove the session from the session store; it merely sets the state to LogoutInProgress.
4. Based on information in the session store, the session is identified as one created by a SAML assertion received from the Identity Provider, smcompany.com.
5. The user's 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's session from all Service Providers associated with that user's 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's session is removed from the session store.
8. The user is finally sent to a logout confirmation page at ahealthco.com.

Terminating both sessions ensures that an unauthorized employee cannot use the existing session to view benefits of the authorized employee.

## Solution 6: WS-Federation Signout

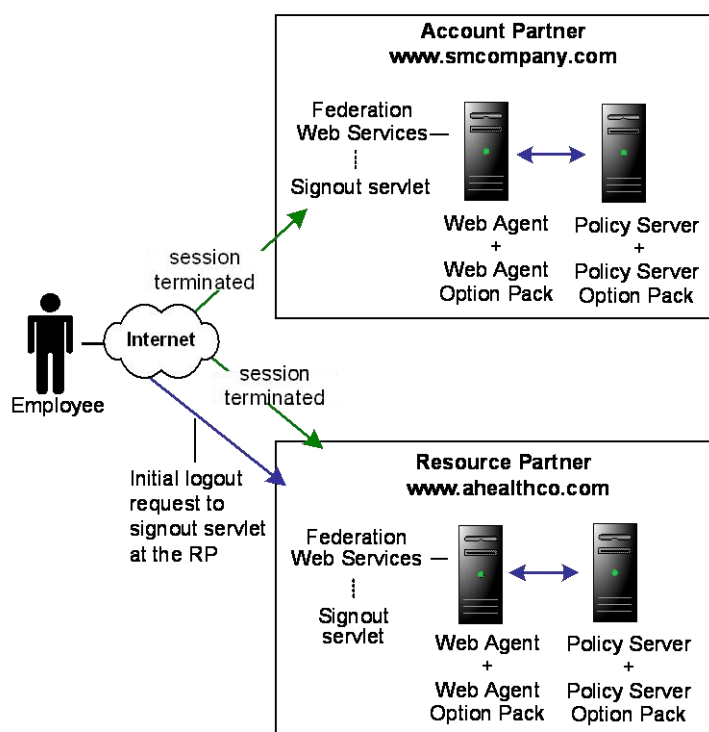
Solution 6 illustrates how SiteMinder Federation Security Services can be employed to solve [Use Case 6: WS-Federation Signout](#) (see page 29).

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 using the Policy Server User Interface 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 SiteMinder Federation Web Services application functions. For 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. An employee performs single sign-on between smcompany.com and ahealthco.com. As a result, smcompany.com places information about ahealthco.com in its session server. Ahealthco.com places information about smcompany.com in its session server.
2. After the employee has finished looking at her health benefits, she clicks a log-out link at the Account Partner, which calls the signout servlet at the Account Partner.
3. The user's session is terminated from the Account Partner's session store and 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 associated with the user session.

5. The user's browser then accesses the signout Cleanup URL at ahealthco.com and the user's session is removed from the session store.
6. The user's browser 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.

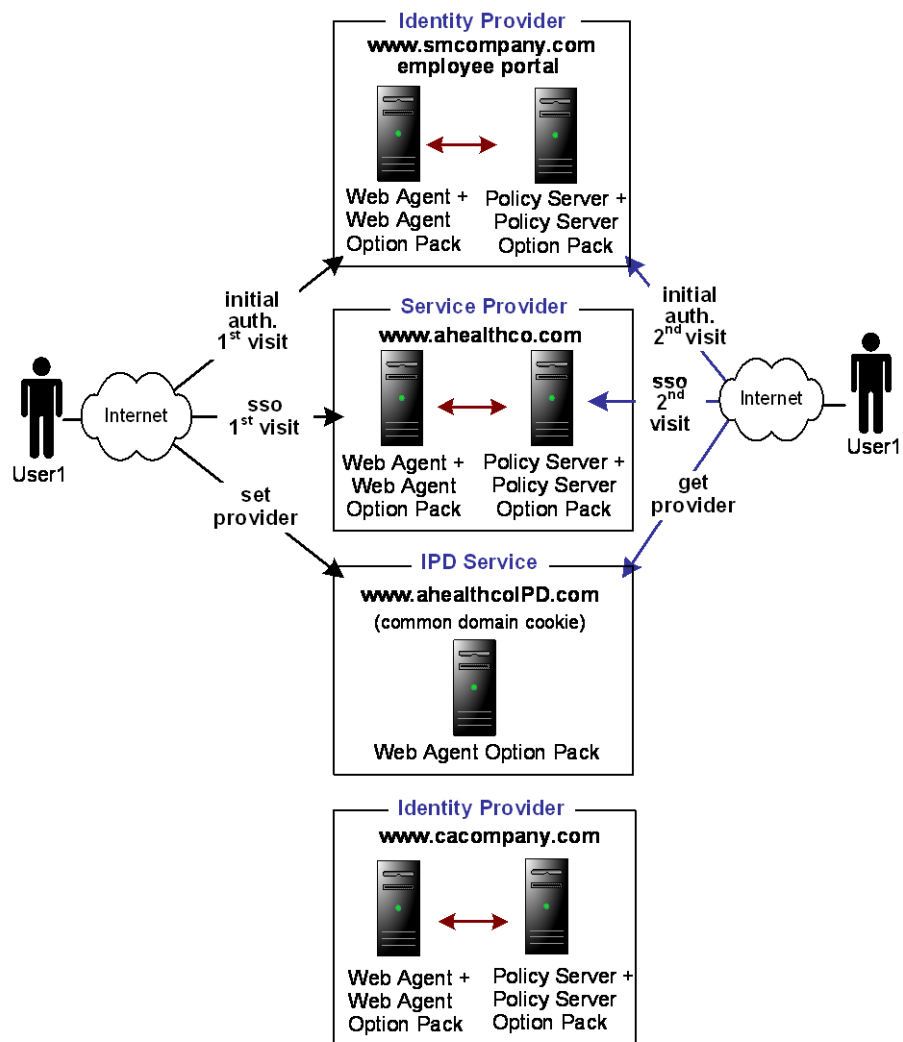
## Solution 7: Identity Provider Discovery Profile (SAML 2.0)

Solution 7 illustrates how SiteMinder Federation Security Services can be employed to solve [Use Case 7: Identity Provider Discovery Profile](#) (see page 30).

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 smfit.com, and has a SAML 2.0 authentication scheme configured for each of these Identity Providers. This enables single sign-on.
- ahealthcoIPD.com is the Identity Provider Discovery Service for ahealthco.com. The Federation Web Services application, installed with the Web Agent Option Pack, provides the IPD service which can read the common domain cookie that 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 SiteMinder federated network for this solution.



**Note:** The SPS federation gateway can replace the Web Agent and Web Agent Option Pack to provide the SiteMinder Federation Web Services application functions. For 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. User 1 initially authenticates at smcompany.com and then signs on to ahealthco.com without having to reauthenticate.

There is an existing agreement 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 his health benefits.
3. User 1 then comes to a site selection page at ahealthco.com. Because a common domain cookie is set for smcompany.com and ahealthco.com is configured to use the IPD service, ahealthco.com knows that the user previously logged into smcompany.com. Therefore, ahealthco.com can make the appropriate links available to the user so that user can go back to smcompany.com to log in.

## Solution 8: Multi-protocol Network

Solution 8 illustrates how SiteMinder Federation Security Services can be employed to solve [Use Case 8: Multi-protocol Support](#) (see page 31).

In this solution:

For User 1:

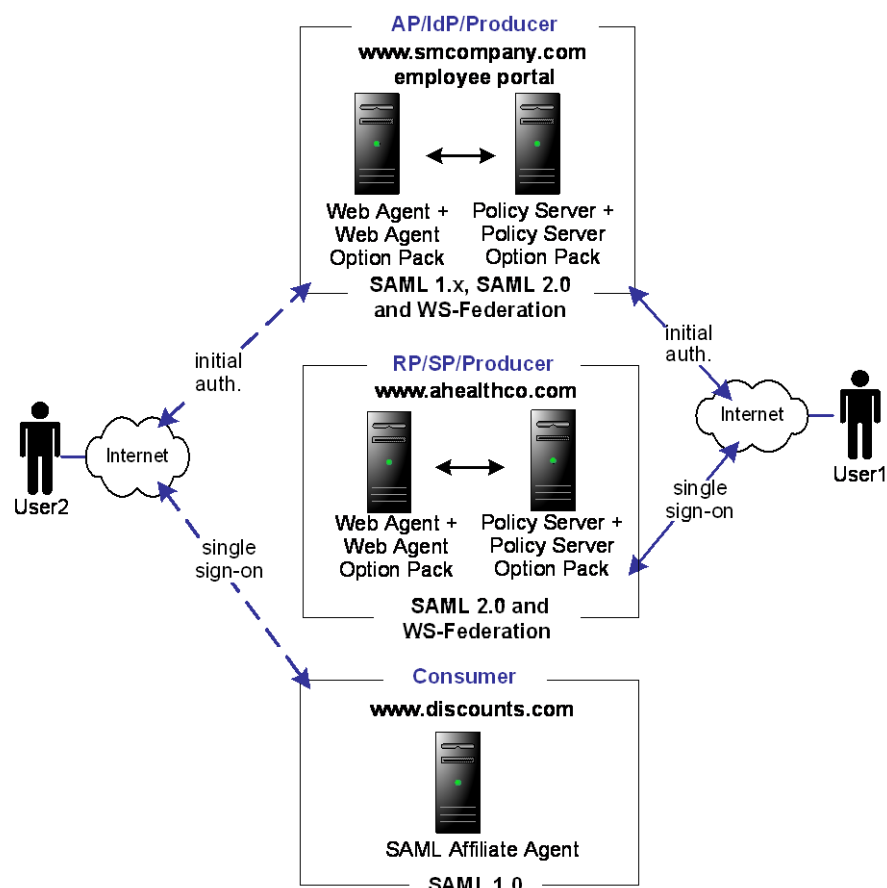
- smcompany.com is the SAML 2.0 Identity Provider for ahealthco.com. The partner, ahealthco.com, is included in an affiliate domain as a SAML 2.0 Service Provider.
- ahealthco.com is where the SAML 2.0 authentication scheme is configured, and where smcompany.com is identified as the Identity Provider.

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. It 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 SiteMinder Federation Web Services application functions. For information about installing and configuring the SPS federation gateway, see the *CA SiteMinder Secure Proxy Server Administration Guide*.

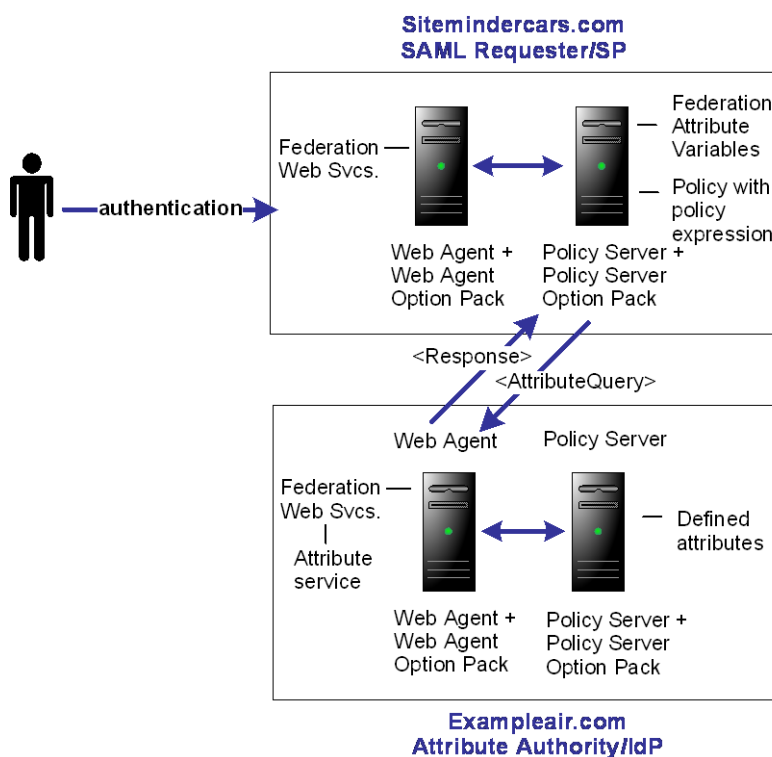
In this multiprotocol solution, smcompany.com can issue a SAML 2.0 assertion for User 1 to access resources at ahealthco.com. Additionally, smcompany.com can issue a SAML 1.0 assertion for User 2 to authenticate at discounts.com. Smcompany.com issues an assertion based on the session cookie that is set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

The SAML Affiliate Agent at discounts.com needs to be configured so that 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.

## 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](http://sitemindercars.com) and [exampleair.com](http://exampleair.com) to solve [Use Case 9: SAML 2.0 User Authorization Based on a User Attribute](#) (see page 31).

**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 SiteMinder Federation Web Services application functions. For information about installing and configuring the SPS federation gateway, see the *CA SiteMinder Secure Proxy Server Administration Guide*.

SiteMinder v6.x is deployed at both sites by installing the Web Agent with the Web Agent Option Pack on a Web server and the Policy Server with the Policy Server Option Pack on another machine. The installations are the same for both sites.

In this solution, sitemindercars.com is a Service Provider acting as a SAML Requester. When a customer logs on at this site, the sequence of events is as follows:

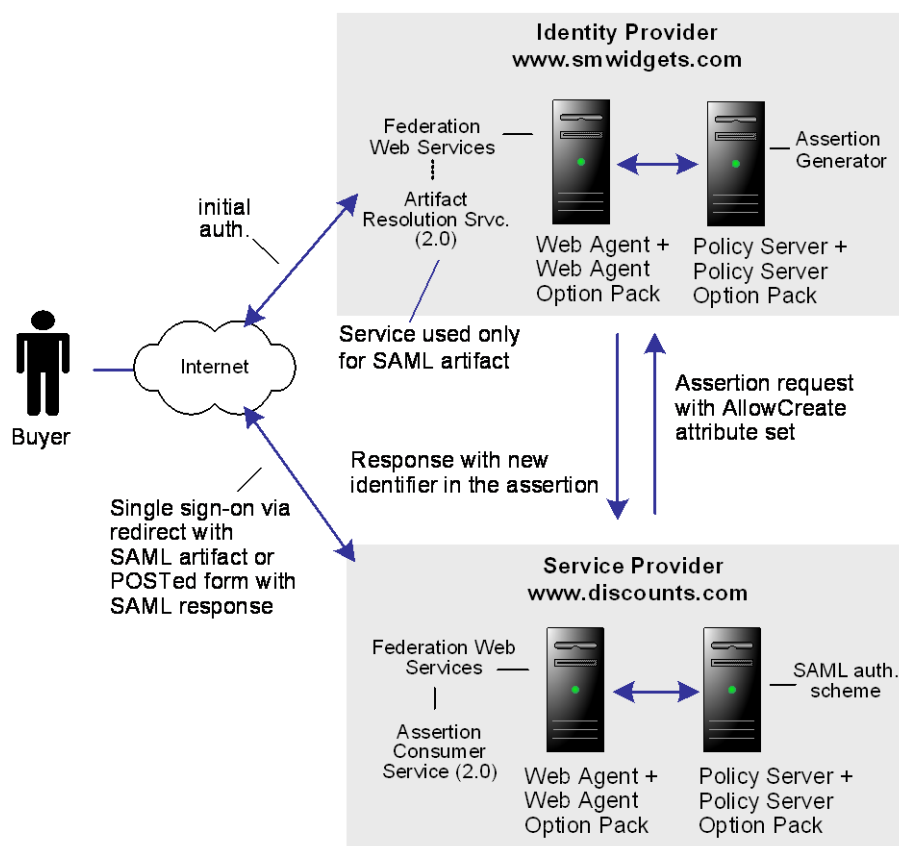
1. The user logs in to sitemindercars.com and the Web Agent provides the authentication.
2. When the user clicks a link to rent a car, the Web Agent first makes a request to the Policy Server at the local site.
3. The Policy Server evaluates the policy expression and determines that there are unresolved federation attribute variables. The Policy Server tries to resolve the variable by looking up the user in the user directory associated with the policy protecting the requested URL.
4. The local Policy Server cannot resolve the user attribute variable. It locates the NameID configuration for the Attribute Authority, Exampleair.com. This is the Attribute Authority where the <AttributeQuery> will be sent.
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.

## Solution 10: Single Sign-on with No User 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 32).

SiteMinder v6.x is deployed at discounts.com and smwidgets.com by installing the Web Agent with the Web Agent Option pack on one machine, and the Policy Server with the Policy Server Option Pack on another machine.

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 SiteMinder Federation Web Services application functions. For information about installing and configuring the SPS federation gateway, see the *CA SiteMinder 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 take him 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 called 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 request may ask for the telephone number as the value of the Name ID.

If the Policy Server cannot find a value for the telephone number attribute, it checks 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 both sides have the Allow/Create feature enabled, the Policy Server generates a new identifier for the user attribute and 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. This form is posted to the Assertion Consumer Service at the Service Provider, which 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, and assuming 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, places it in the user's browser, and redirects the user to the target destination.

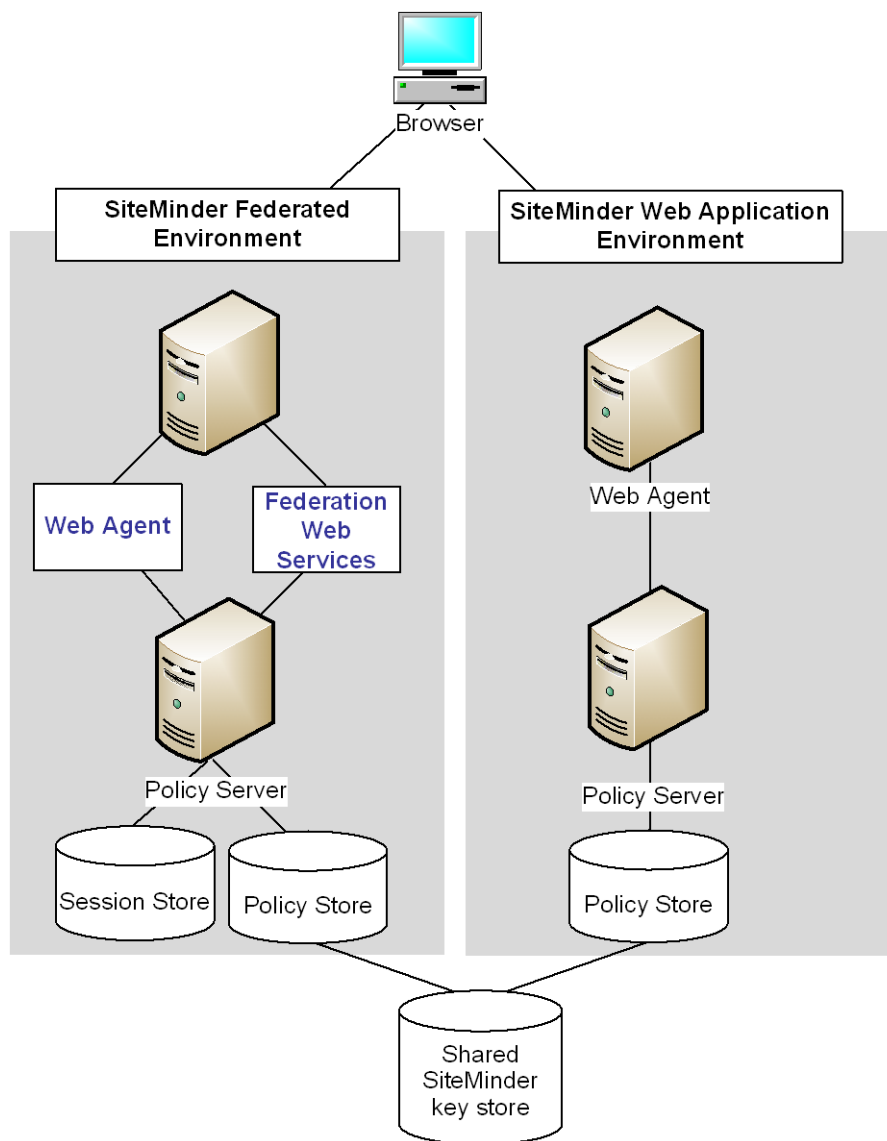
## Solution 11: SAML Artifact SSO Using Security Zones

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

SiteMinder's security zones can be used to eliminate the need for persistent user sessions associated with every request for applications protected by the SAML 1.x or SAML 2.0 artifact profiles.

A security zone is a segment of a single cookie domain, used as a method of partitioning applications to assign different security requirements. Security zones are implemented by Web Agents and can be configured for the producer-side Web Agents that are protecting the requested federated resources.

The following figure illustrates a deployment that uses two different SiteMinder environments at a single producing authority site. 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 SiteMinder Federation Web Services application functions. For information about installing and configuring the SPS federation gateway, see the *CA SiteMinder 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 that Can be 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 and is backwards compatible with SiteMinder 5.x. |

#### Federation Environment

| Agent Configuration Object or Local Configuration File | Trusted Security Zones                              | Cookies that Can be Read by the Web Agent for the Zone                                                              |
|--------------------------------------------------------|-----------------------------------------------------|---------------------------------------------------------------------------------------------------------------------|
| FedWA<br>used by the Web Agent                         | FED--primary zone<br>SM--additionally accepted zone | The FedWA configuration enables the Web Agent to read and write SESSIONSIGNOUT cookies, and read SMSESSION cookies. |
| FedFWS<br>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 ensures that requests to web applications do not incur the performance penalty of calling the session server.

When the Web Agent in the federation environment receives a request, this request is directed to the Authentication URL to establish a user session. This user already has an SMSESSION cookie established by a prior authentication in the Web application environment, but the user has no SESSIONSIGNOUT cookie. The Authentication URL, which authenticates users federating to a partner site, is protected by a persistent user session in the federation environment. This is why the Web Agent in the federation environment writes a SESSIONSIGNOUT cookie with a persistent user session specification and with the same session ID as the SMSESSION cookie.

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.

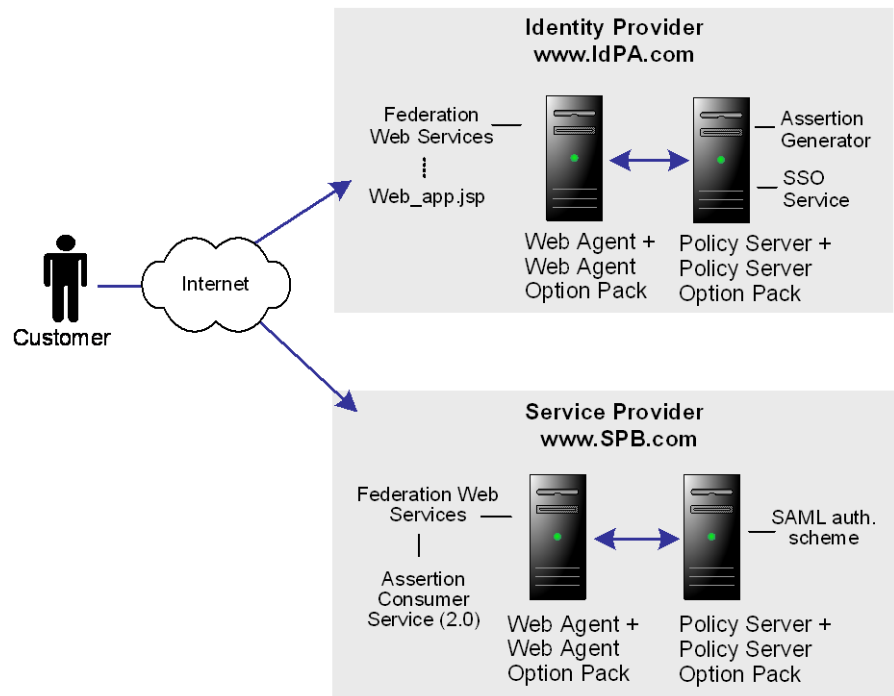
The FWS application in the federation environment reads the SESSIONSIGNOUT cookie. This cookie contains a persistent user session, therefore, a call to the session server is not necessary. The FWS application can successfully process federation requests that requires a persistent user session.

## Solution 12: SAML 2.0 SSO with Attributes from a Web Application

Solution 12 shows how Federation Security Services can be deployed at IdPA.com and SPB.com to solve [Use Case 12](#) (see page 34). This solution is available only for SAML 2.0 deployments.

SiteMinder v6.x is deployed at both sites by installing the Web Agent with the Web Agent Option pack on one machine, and the Policy Server with the Policy Server Option Pack on another machine.

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 SiteMinder Federation Web Services application functions. For information about installing and configuring the SPS federation gateway, see the *CA SiteMinder Secure Proxy Server Administration Guide*.

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

1. At the IdP, the user clicks on web page link and *one* of the following occurs:
  - The user is directed to IdPA.com's Single Sign-on (SSO) service. This service recognizes that the user does not have a session, so the user is redirected to the Authentication URL where he is prompted to log in. After successfully logging in, the user is redirected back to the SSO service. There is an application URL defined for the SSO service, which instructs the SSO service to redirect 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 called SMPORTALSTATE, and then redirects the user (via 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, he is never redirected to the Authentication URL as long as he has a valid session.

2. Prompted by the web application, the user supplies the requested information. These attributes are POSTed to the SSO Service.

**Important!** If the user starts at the SSO Service, and is redirected to the web application with the SMPORTALSTATE query parameter, the web application must POST the SMPORTALSTATE query parameter and the collected attributes back to the SSO Service.

3. The SSO service processes the SAML request. It unpacks the data from the SMPORTALSTATE parameter takes this data along 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, but an Assertion Generator plug-in must be written and configured to add the attributes to the assertion.

5. After the assertion is generated, the user is redirected to the Assertion Consumer Service at the Service Provider, where the assertion is processed.
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 case of SP-initiated single sign-on, the request must arrive at the SSO service directly from the SP, as dictated by SAML specifications. 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, so the user is redirected to the Authentication URL where he is prompted to log in. After successfully logging in, the user is redirected back to the SSO service. There is an application URL defined for the SSO service, which instructs the SSO service to redirect 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 SSO request. The SSO service groups this query data into one query parameter called SMPORTALSTATE, and then redirects the user (via a GET) to the web application.

3. Prompted by the web application, 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.

4. The SSO service processes the SAML request. It unpacks the data from the SMPORTALSTATE parameter takes this data along with the attributes from the web application and passes all the POST data to the Assertion Generator.
5. The assertion is generated with all the attributes and the user is redirected to the Assertion Consumer Service at the Service Provider, where the assertion is processed.

**Note:** An Assertion Generator plug-in must be written and configured to add the attributes to the assertion.

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

**More information**

[Use Case 12: SAML 2.0 SSO Using Attributes from a Web Application](#) (see page 34)

[Configure SSO with Attributes from a Web Application](#) (see page 67)

[Integrate the Assertion Generator Plug-in with SiteMinder \(SAML 2.0/WS-Federation\)](#) (see page 341)

## Configure SSO with Attributes from a Web Application

To configure single sign-on based on attributes from a web application, configure the following

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 or it can simply 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** for the POST to be accepted by the SSO service.

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, when 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 when the user is sent directly to the web application and not initially to the SSO Service. It assumes the user is already authenticated at the Identity Provider.

**Note:** It 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. The Assertion Generator Plug-in is specified in the Advanced tab of the SAML Service Provider Properties dialog.
3. (Optional) If the user is sent directly to the custom web application when they click on a link at the IdP, you do not have to provide a value for the Application URL parameter in the Policy Server User Interface. However, the Assertion Generator plug-in still needs to be written and configured to work with SiteMinder.

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

**More information:**

[Integrate the Assertion Generator Plug-in with SiteMinder \(SAML 2.0/WS-Federation\)](#)  
(see page 341)

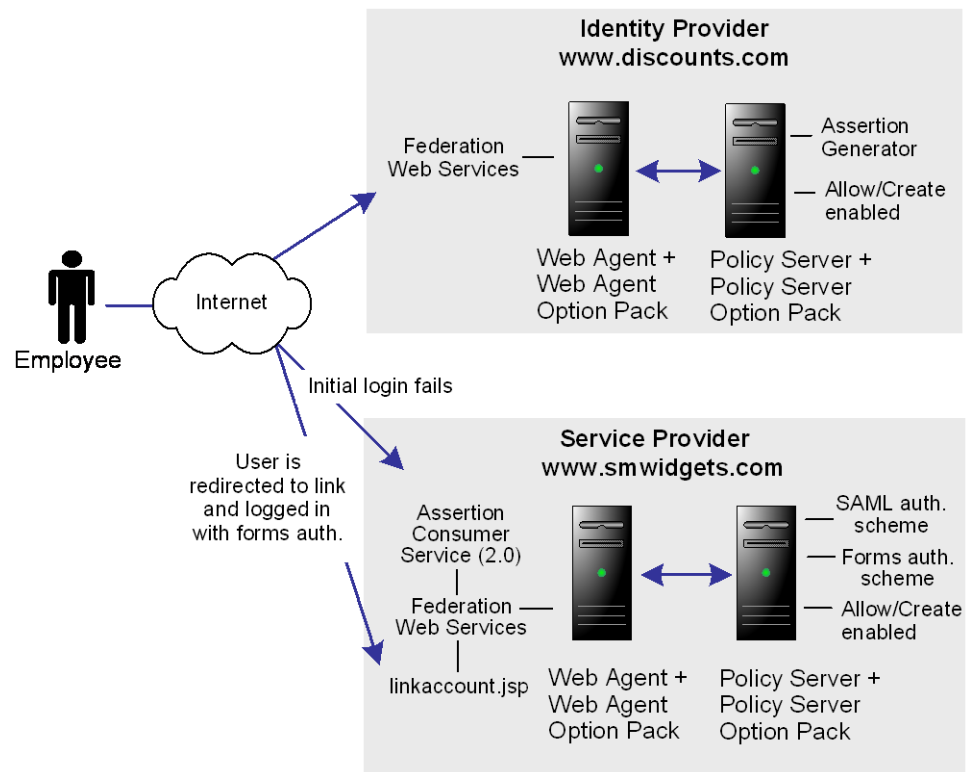
## Solution 13: SAML 2.0 SSO with Dynamic Account Linking at the SP

Solution 13 shows how SiteMinder Federation Security Services can be deployed at IdPA.com and SPB.com to solve [Use Case 13](#) (see page 36).

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

SiteMinder v6.x is deployed at both sites by installing the Web Agent with the Web Agent Option pack on one machine, and the Policy Server with the Policy Server Option Pack on another machine.

The following figure 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 SiteMinder Federation Web Services application functions. For information about installing and configuring the SPS federation gateway, see the *CA SiteMinder Secure Proxy Server Administration Guide*.

**The order of events for this solution is as follows**

1. The employee initially logs in and authenticates at discounts.com. Discounts.com creates an assertion for the employee and posts the assertion (for POST binding), or redirects the user with an artifact (for artifact binding) to the Assertion Consumer Service at smwidgets.com. This assertion includes an attribute called buyerID.
2. The AssertionConsumer Service at smwidgets.com tries to authenticate the user with the SAML authentication scheme, but the employee's buyerID attribute does not map to a local user record so 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 is for this file is `http://sp_home/affwebservices/public/`. Copy the file from this location to a protected realm.

4. This linkaccount.jsp URL is protected by a Web Agent that authenticates the local user with the forms authentication scheme. After a successful authentication, a SiteMinder session at smwidgets.com is established and an SMSESSION cookie is placed in the employee's browser.
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 employee's browser presents the SMSESSION cookie along with the assertion to the Assertion Consumer Service.
7. The Assertion Consumer Service extracts the NameID from the assertion and inserts this value into a newly created buyerID attribute in the employee's existing user record. The Assertion Consumer Service knows which user record to map because the user is identified by the UserDN in the SMSESSION cookie.

The Search Specification that is 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 user's attribute is mapped, the SAML authentication scheme authenticates the user based on the assertion and establishes a new user session.

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

## Configure SAML 2.0 SSO with Dynamic Account Linking at the SP

You need to 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. This URL is protected by an authentication scheme that prompts the user to log in so a SiteMinder session is created.
- Post Preservation at the Web Agent--must be enabled at the Service Provider Web Agent
- Search Specification-- indicates which attribute will be replaced by the NameID from the assertion.

**To enable dynamic account linking for POST or Artifact single sign-on, configure the following at the Service Provider**

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 that 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. Using a scheme that supports POST preservation is necessary because 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. The SAML response POST data needs to be preserved 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 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. You must direct the user only to this file.

The redirect URL is part of the SAML 2.0 authentication scheme setup at the Service Provider, specifically, in the Advanced tab of the SAML Auth Properties dialog.

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 will be configured as a protected realm.

**Mode**

Http POST

5. Configure a Search Specification in the Users tab of the SAML Auth Scheme Properties dialog. For example, if `buyerID` is going to be replaced by the Name ID from the assertion, the Search Specification would be `buyerID=%s`.

**More information:**

[Use Case 13: SSO with Dynamic Account Linking at the SP](#) (see page 36)

[Solution 13: SAML 2.0 SSO with Dynamic Account Linking at the SP](#) (see page 68)

[Allow the Identity Provider to Assign a Value for the NameID](#) (see page 319)

[Permit the Creation of a Name Identifier for SSO](#) (see page 368)

[Configure User Disambiguation for User Look Ups](#) (see page 360)

[Locate User Records for Authentication](#) (see page 436)

## Federation Security Services Concepts

Before implementing SiteMinder Federation Security Services, you may find it helpful to have a understanding of some basic security concepts, such as different SAML protocol versions, WS-Federation, entities within a federated network, and user mapping.

### Security Assertion Markup Language (SAML)

The Security Assertion Markup Language (SAML) is a standard developed by the Organization for the Advancement of Structured Information Standards (OASIS). It is an industry standard that defines an XML framework for exchanging authentication and authorization information.

SAML defines assertions as a means to pass security information about users between entities. SAML assertions are XML documents that contain information about a specific subject, such as a user. An assertion can contain several different internal statements about authentication, authorization, and attributes.

SAML defines two browser-based protocols that specify how SAML assertions are passed between partners to facilitate single sign-on.

The profiles are:

- Browser/artifact profile—defines a SAML artifact as a reference to a SAML assertion.
- Browser/POST profile—returns a response that contains an assertion

**Note:** For SAML 2.0, the artifact and POST profiles are referred to as HTTP bindings.

For SAML specifications and background documentation as well as information on SAML profiles, go to site for the [Organization for the Advancement of Structured Information Standards \(Oasis\)](#).

## WS-Federation

Active Directory Federation Services (ADFS) is Microsoft's Web Services-based solution for federation and single sign-on (SSO). ADFS runs on Windows Server 2003 R2 and accomplishes SSO by letting partners securely share a user's 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 the organization's Web-based applications.

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 as well as information on ADFS profiles, go to the [Microsoft web site](#).

## Entities in a Federated Network

In a federated network, there is an entity that generates SAML assertions. Assertions contain information about a user whose identity is maintained locally at the site that generates them. There is another entity that uses the SAML assertions to authenticate a user and to establish a session for the user.



Depending on the protocol, these two entities are named differently, but the functions they serve are the same.

| Protocol         | Generates Assertions    | Consumes Assertions   |
|------------------|-------------------------|-----------------------|
| SAML 1.0 and 1.1 | Producer                | Consumer              |
| SAML 2.0         | Identity Provider (IdP) | Service Provider (SP) |
| WS-Federation    | Account Partner (AP)    | Resource Partner (RP) |

A site may be both a producing authority (producer/IdP/AP) and a consuming authority (consumer/SP/RP).

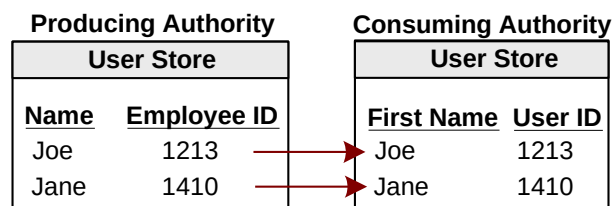
## User Mapping

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 a producing authority to local users at a consuming authority.

There are two types of mapping:

- One-to-one mapping maps a unique remote user directory entry at the producing authority to a unique user entry at the consuming authority.

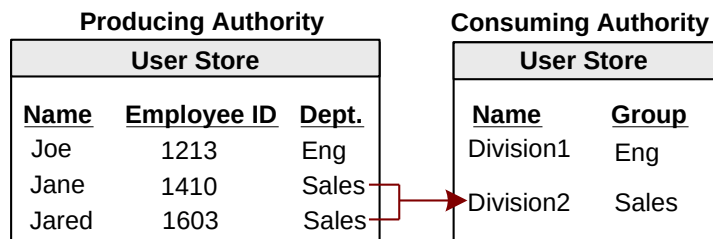
One-to-one mapping is often referred to as account linking, as it links an account at a producing authority site to an account at a consuming authority, as shown in the following illustration:



- 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 producing authority to be mapped to one user record or profile at a consuming authority. An administrator at the consuming authority 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:



## 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 because the SAML assertion must be stored in the session store at 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 is later retrieved by the consuming site.
- 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 can slow down performance because of the calls to the session store to retrieve assertions or handle log-off requests.

For producer-side applications protected by a Web Agent, security zones can eliminate the need for a persistent user session. A security zone is a segment of a single cookie domain, used as a method of partitioning 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, single sign-on is determined by the configured trust relationship.

Security zones are a SiteMinder single sign-on feature and are implemented by SiteMinder Web Agents.

**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, you enter values for the following Web Agent parameters:

- **SSOZoneName**--identifies a single sign-on security zone. The zone name gets added to the cookie domain name so you know which are associated with which domains.
- **SSOTrustedZone**--ordered list of trusted security zone names. When you define zones and trusted zone lists, it determines the cookies that the Web Agent is able to read and/or write.

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

To find more information about security zones, see the *SiteMinder Web Agent Guide*.

## SAML Profiles Supported by SiteMinder

SiteMinder's 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

## Benefits of SiteMinder Federation Security Services

SiteMinder Federation Security Services supports:

- **Secure profile sharing**—The ability to exchange user profile information with partners in a secure manner.
- **Flexible attribute sharing**--You control which user attributes to share with which partners.
- **Flexible user mapping**—The ability to establish a one-to-one or n-to-one relationship between remote producer/IdP and local consumer/SP user accounts.
- **Cross-domain single sign-on**—A user session can be established in a different domain from the domain where the user was initially authenticated without requiring the user to sign on multiple times. Additionally, for SAML 2.0, single sign-on can be supported in a multi-SAML protocol cross-domain environment.

- Enhanced Client or Proxy profile (ECP) for single sign-on (SAML 2.0)--The ECP profile determines how an enhanced client (browser or user agent) or HTTP proxy wireless access protocol (WAP) gateway can communicate with a Service Provider and an Identity Provider. An ECP knows how to contact the appropriate Identity Provider associated with a user, allowing a Service Provider to make an authentication request without knowledge of the Identity Provider.
- Cross-domain single logout/signout--A user session can be terminated across different domains, regardless of whether the logout was initiated at the producing authority or the consuming authority.
- Identity Provider Discovery Profile (SAML 2.0)--You control which Identity Provider a user relies on for obtaining an assertion by using the SiteMinder Identity Provider Discovery Service, which stores Identity Provider information in a common domain cookie.
- Policy-based access control--Once a user session is established based on user information received from a partner, all the power of SiteMinder is available to control access to resources through a centralized policy administration model.
- Rich session models (SAML Affiliate Agent only)--If the SAML Affiliate Agent is acting as the consumer, you can configure separate portal and affiliate sessions, a single session at the portal, or a shared session that provides single sign-on as well as single sign-off. The SAML Affiliate Agent only supports SAML 1.0.  
**Note:** These session models are not applicable if the SAML credential collector is the consumer.
- Notifications (SAML Affiliate Agent only)--If the SAML Affiliate Agent is acting as the consumer, it can notify the SAML producer when the user accesses specific resources at the affiliate site.
- Interoperability through the use of open standards--Standards facilitate interoperability across heterogeneous environments. SiteMinder Federation Security Services supports the following standards:
  - SAML, to provide the structure for sharing security data
    - HTTP, for communication between Web browsers and servers
    - SSL, for encrypting security data passed between partners
    - SOAP, to provide an envelope for the SAML messages exchanged between a producer and consumer
  - Policy-based model--All these benefits are provided using a policy-based model that does not require any code to be written.

## 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 a producing authority site, as well as collecting assertions at a consuming authority site.
- 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 request for a SAML assertion is made, the Web Agent invokes the SAML assertion generator, which creates an assertion based on the user session and information configured in the policy store.

The assertion is then handled according to the authentication profile or binding configured, as follows:

- SAML artifact profile/binding--assertion is placed in the SiteMinder session server and a reference to the assertion is returned to the Web Agent in the form of a SAML artifact.
- SAML POST profile/binding--assertion is returned via the user's browser as a SAML response embedded in a HTTP form.

The Web Agent is responsible for sending the SAML artifact, SAML response, or WS-Federation security token response message to the site that will consume the assertion accordance with the SAML profile or binding. At the consumer/SP site, a client, such as the SAML Affiliate Agent, the SAML 1.x credential collector or the SAML 2.0 assertion consumer, must be available to process the SAML artifact or response message.

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 and Policy Server Option Pack. After installing the Policy Server and Policy Server Option Pack, the administrator can use the Policy Server User Interface to define and configure affiliates.

## 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 and Policy Server Option Pack. After installing the Policy Server and Policy Server Option Pack, the Account Partner administrator can use the Policy Server User Interface 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 a producing authority to local users at the consuming authority. 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 and Policy Server Option Pack. After installation, the administrator can use the Policy Server User Interface 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

The Federation Web Services (FWS) application is installed with the Web Agent Option Pack on a server that has a connection to a SiteMinder Policy Server. The Federation Web Services and the SiteMinder Web Agent support the following protocols:

- SAML browser artifact protocol
- SAML POST profile protocol
- WS-Federation Passive Requestor profile protocol

## SAML Browser Artifact Protocol

For the SAML browser artifact protocol, the Federation Web Services application includes the following services:

- Assertion Retrieval Service (SAML 1.x)--A producer site component. This service handles a SAML request for the assertion that corresponds to a SAML artifact by retrieving the assertion from the SiteMinder session server. The assertion retrieval request and response behavior is defined by the SAML specification.  
**Note:** The assertion retrieval service is used only by the SAML artifact profile, not by the SAML POST profile.
- Session Synchronization (SAML 1.x)--A producer site component that validates and terminates sessions for the SAML Affiliate Agent (A SiteMinder value-added service, supported by a standards-based SOAP RPC mechanism)
- Notification Alert (SAML 1.x)--A producer site component that logs resource access notification events for the SAML Affiliate Agent (A SiteMinder value-added service, supported by a standards-based SOAP RPC mechanism)
- SAML Credential Collector (SAML 1.x)--A consumer site 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 user's browser.
- Intersite Transfer Service (SAML 1.x)--For SAML POST profile, a producer site component that transfers a user from the producer site to a consumer site. For SAML artifact profile, the same function is performed by the Web Agent, which acts as the Intersite Transfer Service.

## SAML POST Profile Protocol

For SAML POST Profile protocol, the Federation Web Services application includes the following services:

- Artifact Resolution Service (SAML 2.0)--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 server at the Identity Provider. This is a SiteMinder-specific service.  
**Note:** The artifact resolution service is used only by the HTTP-artifact binding.
- 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 user's browser.  
**Note:** The Assertion Consumer Service will accept an AuthnRequest with an AssertionConsumerServiceIndex value of 0. All other values for this setting will be denied.



- **AuthnRequest Service (SAML 2.0)**--This service, a SiteMinder-specific service, is a servlet deployed as part of the Federation Web Services application for SAML 2.0. It implements processing for a Service Provider to 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 user's browser to the single sign-on service at the Identity Provider. The AuthnRequest service is used for single sign-on using the POST or artifact binding.

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

- **Single Sign-on Service (SAML 2.0)**--This service implements processing for an Identity Provider to process an AuthnRequest message and gather the necessary SP configuration information to authenticate the user, redirect the user to the Web Agent to authenticate, and invokes the assertion generator to obtain an assertion that is passed back to the Service Provider.
- **Single Logout Service (SAML 2.0)**--This service implements processing of single logout functionality, which can be initiated by an Identity Provider or a Service Provider.
- **Identity Provider Discovery Service** -- 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 Protocol

For WS-Federation Passive Requestor profile protocol, the Federation Web Services application includes the following 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 user's browser.
- **Single Sign-on Service**--Enables processing for an Account Partner to process a wsignin WS-Federation message and gather the necessary Resource Partner configuration information to authenticate the user, redirect the user to the Web Agent to authenticate, and invokes the assertion generator to obtain an assertion that is passed back to the Resource Partner.
- **Signout Service**--Implements processing of single logout functionality by way of a signout servlet. Signout can be initiated by an Account Partner or a Resource Partner.

## SAML Affiliate Agent

The SAML Affiliate Agent enables businesses using the SiteMinder 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 that provides single sign-on and session management capabilities to a consumer site that does not use the SiteMinder Policy Server and Web Agent. The consumer site, 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 affiliate policies configured at the portal, information can be passed to the affiliate and set as cookies or header variables for the affiliate web server.

For complete 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.

**More information:**

[Federation Web Services](#) (see page 79)

## Debugging Features

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

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

## APIs for Federation Security Services

There are several APIs that 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 Federation Security Services. These include:

- C structures and Perl packages for Federation Security Services objects such as Affiliates, Service Providers, Resource/Account Partners, and Affiliate Domains. These objects are required to generate SAML assertions.
- 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 Scripting 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, you can integrate the custom plug-in into SiteMinder to further process and manipulate the SAML 2.0 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 will contain multibyte characters, you need to set your operating system's LANG setting to UTF-8 format, as follows:

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

For example, for Japanese, the entry would be:

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

## Federation Security Services Process Flow

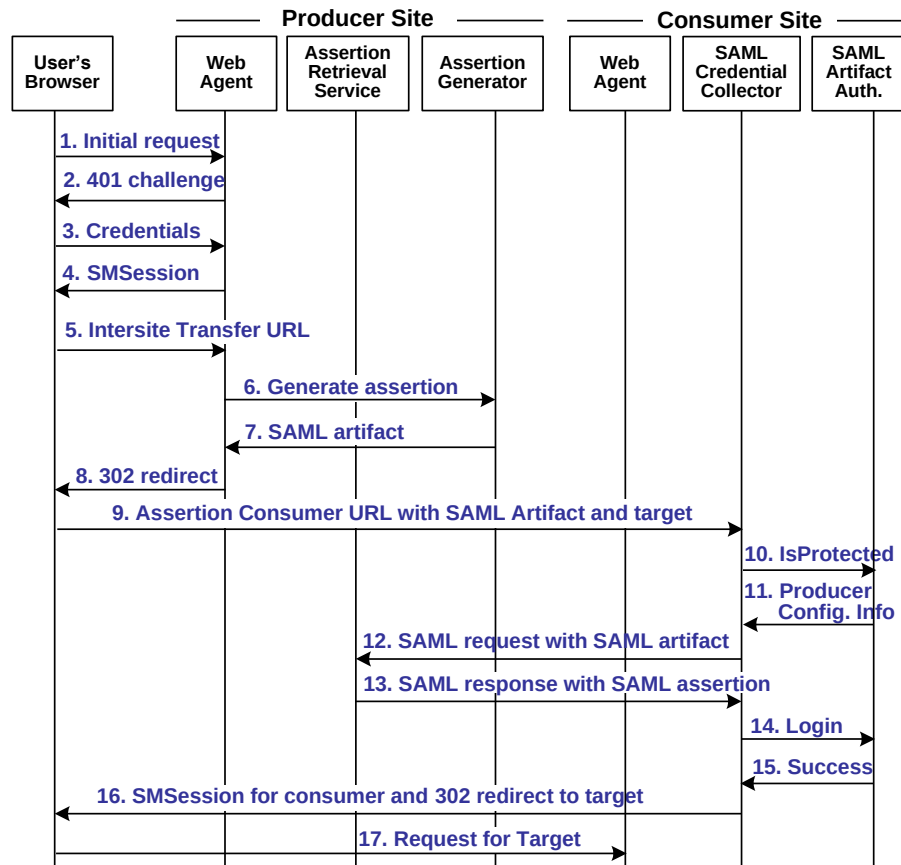
This section shows the detailed flow between the components that comprise the SiteMinder Federated Security Services. It assumes that the reader has some knowledge of SiteMinder interactions between a Web Agent and Policy Server.

### Flow Diagram for SSO Using SAML 1.x Artifact Authentication

The illustration that follows shows the detailed flow between a user's browser and the Federation Security Service components deployed 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 SiteMinder 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 *CA SiteMinder 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 user name and password to the Web Agent.
4. The Web Agent issues a SiteMinder SMSESSION cookie to the user's browser 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 SiteMinder session server 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 with the SAML artifact and the target URL as query parameters.
9. The user's browser makes a request to the SAML credential collector at the consumer site. This is known as the assertion consumer URL.
10. The SAML credential collector handles the assertion consumer 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 server and 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.

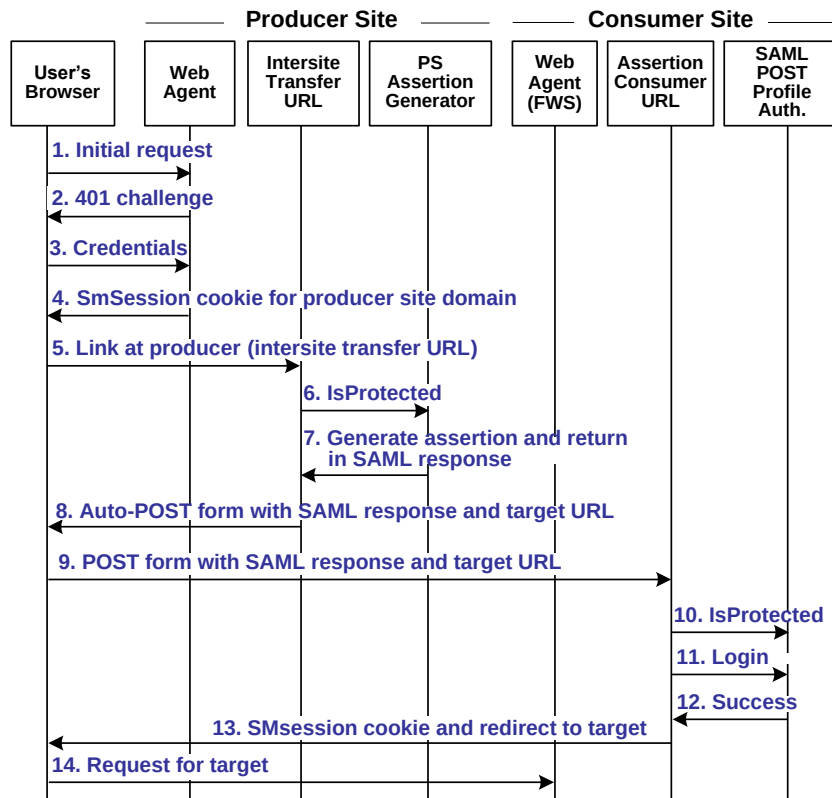
15. The SAML artifact authentication scheme validates the SAML assertion. It looks up the user record for the user based on the user mapping information configured for the SAML authentication scheme, and returns a success reply. If the SAML assertion is not valid or a user record can not be located, a failure is returned.
16. If a success reply is returned, the SAML credential collector issues a SiteMinder SMSESSION cookie to the user's browser for the consumer site domain. It also issues a 302 redirect to the target URL. If failure is returned from the SAML artifact authentication scheme, the SAML credential collector issue a 302 redirect to a no access URL.
17. The user's browser makes a request to the target URL, which is protected by the Web Agent at the consumer.

## Flow Diagram for SSO Using SAML 1.x POST Profile Authentication

The illustration that follows shows the detailed flow between a user's browser and the Federation Security Service components deployed at producer and consumer sites. This set-up enables single sign-on between the sites. SAML POST profile is the authentication method and the 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 POST Profile follows.



**Note:** The SPS federation gateway can replace the Web Agent and Web Agent Option Pack to provide the SiteMinder 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 *CA SiteMinder Secure Proxy Server Administration Guide*.

The sequence of events is as follows:

1. User requests a local page at the producer, which is protected by the Web Agent.
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 username 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 clicks a link at the producer's local page to visit the consumer. The link looks like it goes to the consumer site but it actually goes to the intersite transfer URL, which 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, 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 and sends the form to the user's browser.
9. The user's browser automatically posts the HTML form to the Assertion Consumer URL at the consumer site. This URL was read from the SAML response message sent by the intersite transfer URL service.
10. The assertion consumer URL 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 assertion consumer URL makes a login call for the requested target resource to the SAML POST profile authentication scheme, passing the assertion as credentials.
12. If login succeeds, the assertion consumer URL generates an SMSESSION cookie for the consumer site domain.
13. The SMSESSION cookie is placed in the user's browser, and the assertion consumer URL redirects the user to the target resource.
14. The browser requests the target resource, which is protected by the consumer site Web Agent. Because the browser has an SMSESSION cookie for the consumer domain, the Web Agent does not challenge the user.

## Flow Diagram for SSO Using SAML 2.0 Authentication with Artifact Binding

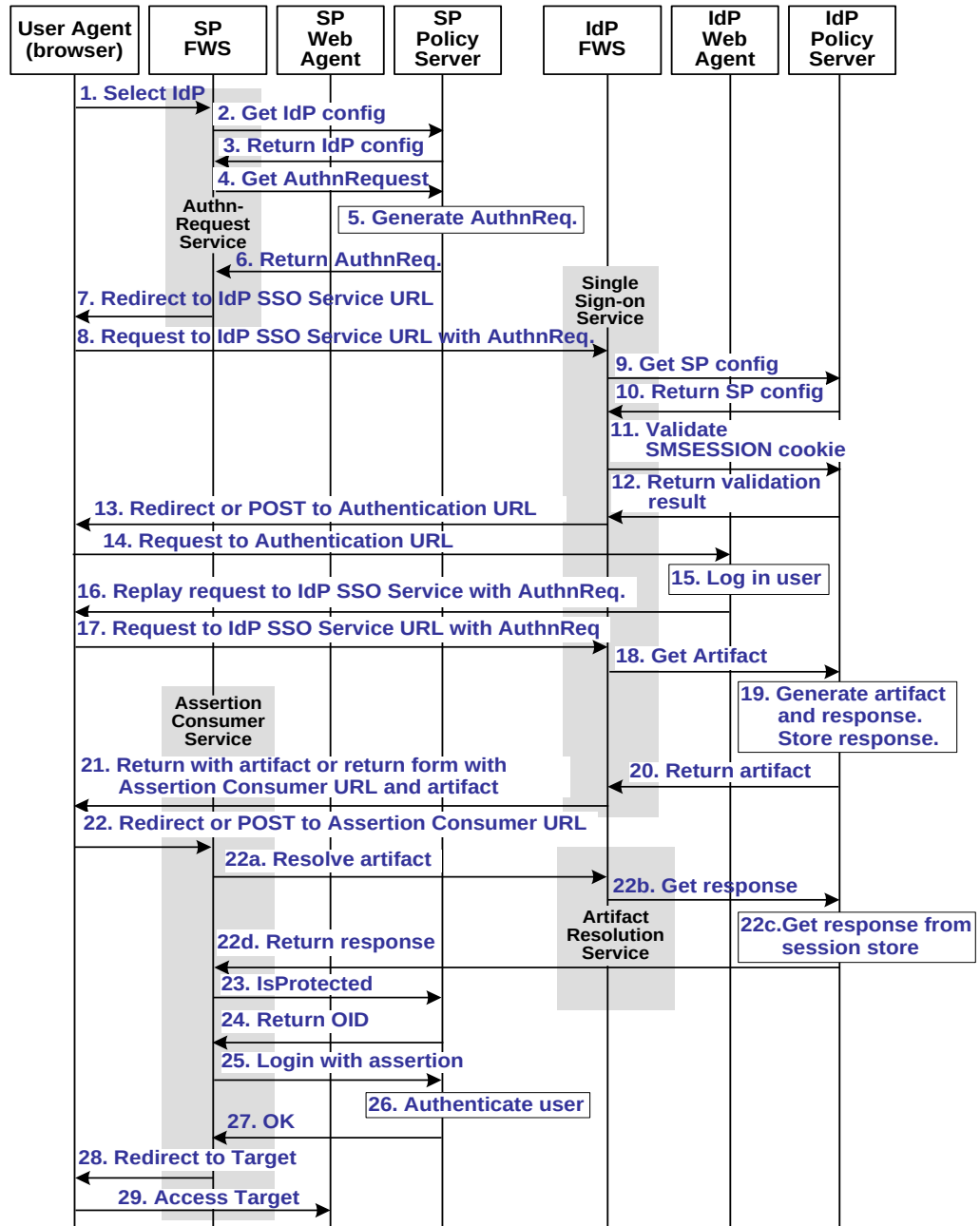
The illustration that follows shows the detailed flow between a user's browser and the Federation Security Service components deployed at the Identity Provider and Service Provider. This set-up enables single sign-on between the sites and uses the SAML 2.0 authentication scheme with artifact binding as the authentication method.

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-Artifact Binding follows.



**Note:** The SPS federation gateway can replace the Web Agent and Web Agent Option Pack to provide the SiteMinder 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 *CA SiteMinder 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 may cache the configuration information.
4. SP FWS requests an AuthnRequest message from the local Policy Server via 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, which is obtained from the configuration information with 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 that FWS may cache the configuration information.
11. IdP FWS gets an SMSESSION cookie for this IdP's domain 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 obtained from the configuration information. 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 is protected by the IdP Web Agent.
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, passing the AuthnRequest via an authorization call to the realm obtained from the configuration information.

19. The Policy Server generates the artifact and the corresponding response message based on the configuration information from the Service Provider. It 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 redirects the browser to the Assertion Consumer URL at the SP with the URL-encoded artifact as a URL parameter, or it 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 read by the browser. The Assertion Consumer URL is obtained from the configuration information.

**Note:** If the assertion generator indicates that the authentication level for the current session 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. It then obtains the SOAP endpoint of the Artifact Resolution Service from the IdP configuration information. The configuration data is retrieved by the source ID, which is part of the artifact. After obtaining the SOAP endpoint, the SP FWS makes a back-channel call to the IdP FWS Artifact Resolution service to resolve the artifact into a response message.

b. The IdP FWS requests the response message from the local Policy Server. The message 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 server 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, 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, which 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 via a login call with 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, places it in the user's browser and redirects the user to the target URL, which is obtained from the configuration information.
- If login fails, the SP FWS redirects the user to a No Access URL.
29. The user's browser requests the target URL, which is protected by the Web Agent at the SP. Because the user's 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

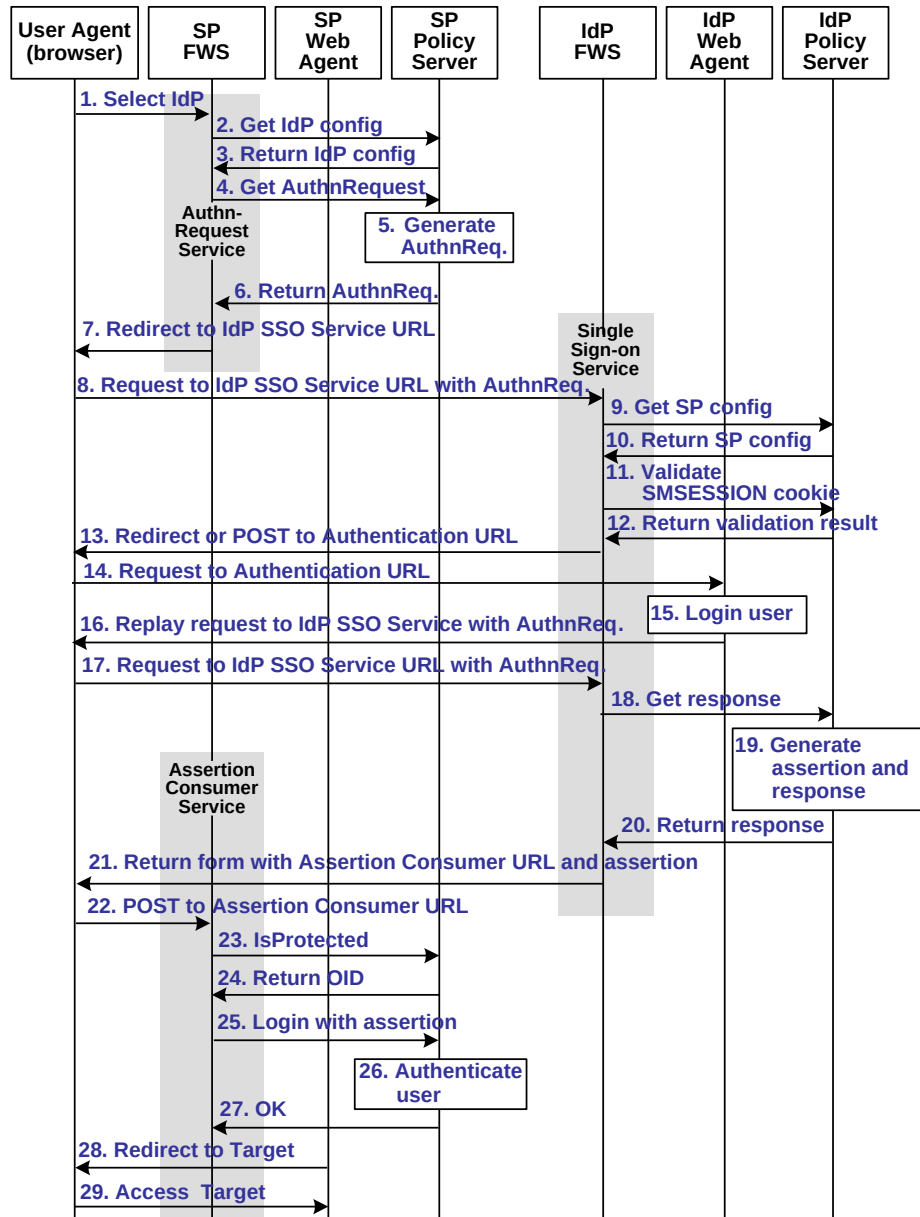
The illustration that follows shows the detailed flow between a user's browser and the Federation Security Service components deployed at an Identity Provider (IdP) and Service Provider (SP) sites. This set-up enables single sign-on between the sites, using SAML 2.0 POST binding as the method of obtaining 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 SiteMinder 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 *CA SiteMinder 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 may cache this configuration information.
4. SP FWS requests an AuthnRequest message from the local Policy Server via 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 that FWS may cache the configuration information.
11. IdP FWS gets an SMSESSION cookie for this IdP's domain 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 obtained from the configuration information. If the SMSESSION cookie is valid, the IDP FWS skips to 18.
14. The browser requests the Authentication URL, which is protected by the IdP Web Agent.
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 request a SAML 2.0 assertion from the Policy Server, passing the AuthnRequest via an authorize call to the realm obtained from the configuration information.

19. The Policy Server generates an assertion 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, which 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 returns an indication 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 user's 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, 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, which 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 via a login call with 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, places it in the user's browser and redirects the user to the target URL, which is obtained from the configuration information.  
  
If login fails, the SP FWS redirects the user to a No Access URL.
29. The user's browser requests to the target URL, which is protected by the Web Agent at the SP. Because the user's browser has an SMSESSION cookie, the Web Agent does not challenge the user.

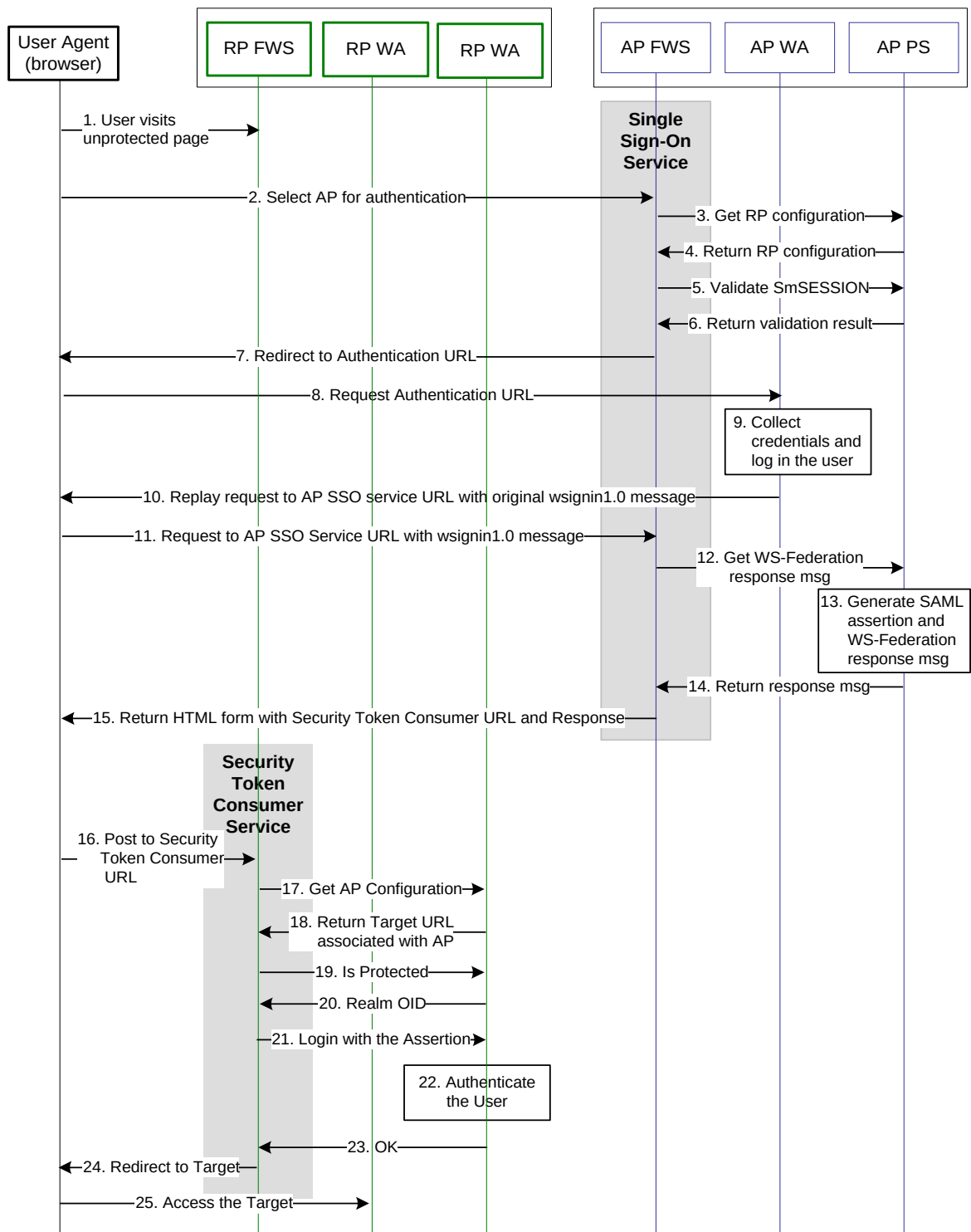


## Flow Diagram for WS-Federation SSO Initiated at the Resource Partner

The illustration that follows shows the detailed flow between a user's browser and the Federation Security Service components deployed 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:

- 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 SiteMinder 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 *CA SiteMinder 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 actually points to the Single Sign-on Service at the AP, and must contain the Provider ID of the RP and may 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 that the FWS may 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 is protected by the AP Web Agent (WA).
9. The AP WA authenticates the user, sets the SMSESSION cookie and 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 via an authorize call to the realm obtained from the configuration information.
13. The Policy Server generates a SAML1.1 assertion based on the configuration information for the RP, signs it, and returns the assertion 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:

- 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 is used as Security Token Consumer URL, only if Security Token Consumer URL config setting is not specified in RP configuration information. For security reasons, the Security Token Consumer URL setting in the RP configuration information takes precedence over the value specified for the wreply parameter.

**Note:** If the assertion generator indicates that the current session's authentication 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 received from local Policy Server. If target resource is not specified as part of AP configuration, and the wctx parameter is found in the POST data, its value is used as 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 via a login call with the <RequestSecurityTokenResponse> message as credentials and the realm OID obtained from the isProtected call.

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 a SMSESSION cookie for the RP domain, places it in the user's browser and redirects the user to the Target URL obtained from the configuration information or to the wctx POST data. If login fails the RP FWS redirects the user to a No Access URL.

25. The user agent requests the Target URL that is protected by the Web Agent at the RP. Since the user's browser has a SMSESSION cookie for the RP domain, the Web Agent does not have to challenge the user.

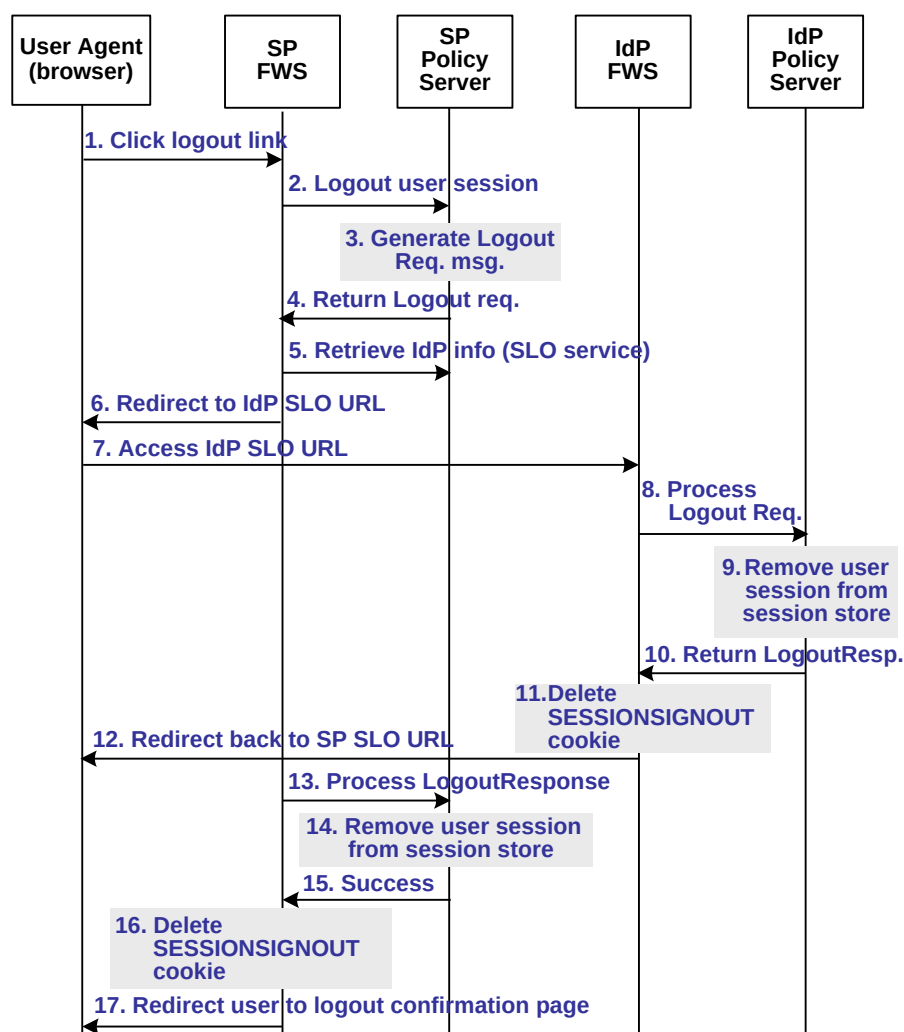
## WS-Federation SSO Initiated at the Account Partner

Single sign-on 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 on any link, the web browser requests the AP SSO Service URL and the rest of the processing is same as specified in in the RP-initiated use case

## Flow Diagram for SAML 2.0 Single Logout

The illustration that follows shows the detailed flow for a single logout request between a user's browser and the Federation Security Service components deployed at an Identity Provider (IdP) and Service Provider (SP) sites. This set-up enables single logout for all entities that have a session with a particular user.

The following diagram 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 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 his global session. The user's browser accesses the Single Logout servlet at the SP.  
  
SP FWS renames the SMSESSION cookie to SESSIONSIGNOUT to invalidate the user's current session.
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 was created based on the SAML assertion received from an IdP. It generates a LogoutRequest request to invalidate the user's session at the IdP.
4. The Policy Server returns a LogoutRequest request to SP FWS. It also returns the IdP's Provider ID and provider type.
5. SP FWS retrieves the IdP's provider configuration data, 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 query parameter.
7. User's browser 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 then tries to invalidate the user's session at all SPs specified in the session store for that user session, with the exception of 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's 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 SP's provider ID and provider type. It also informs FWS that user session is removed from 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 query parameter. The single logout service is part of the SP FWS application.

The user's browser accesses SP's SLO service, which processes the signed LogoutResponse message.

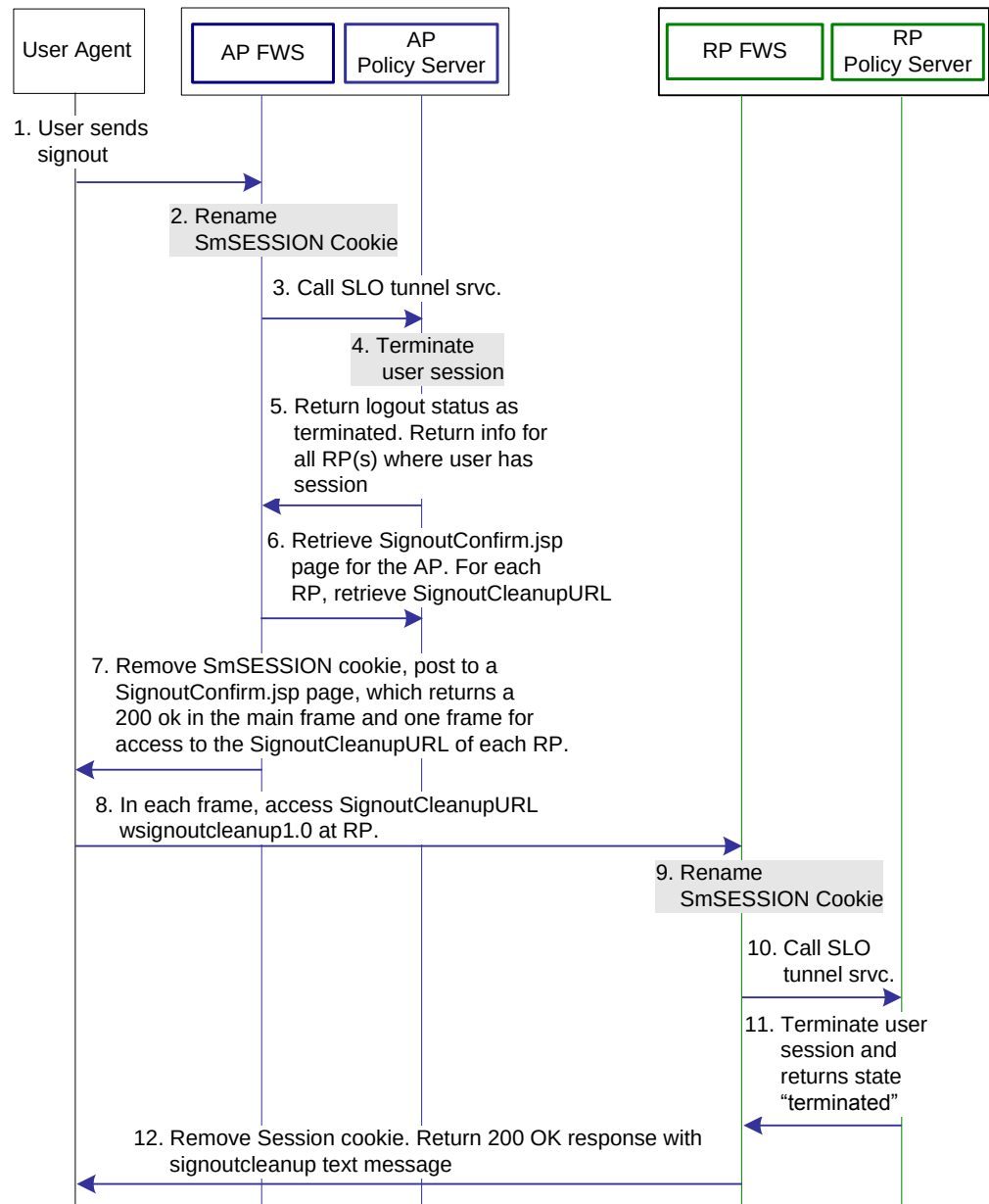
If the LogoutResponse message contains non-SUCCESS return code, FWS issues a SIGNOUTFAILURE cookie, and a base 64-encoded Partner ID is appended to the cookie value. If there are multiple IDs in the cookie, they are separated by a space character.

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 along with the SP provider 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 illustration that follows shows the detailed flow for a signout request between a user's browser and the Federation Security Service components deployed at an Account Partner (AP) and Resource Partner sites. 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 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:

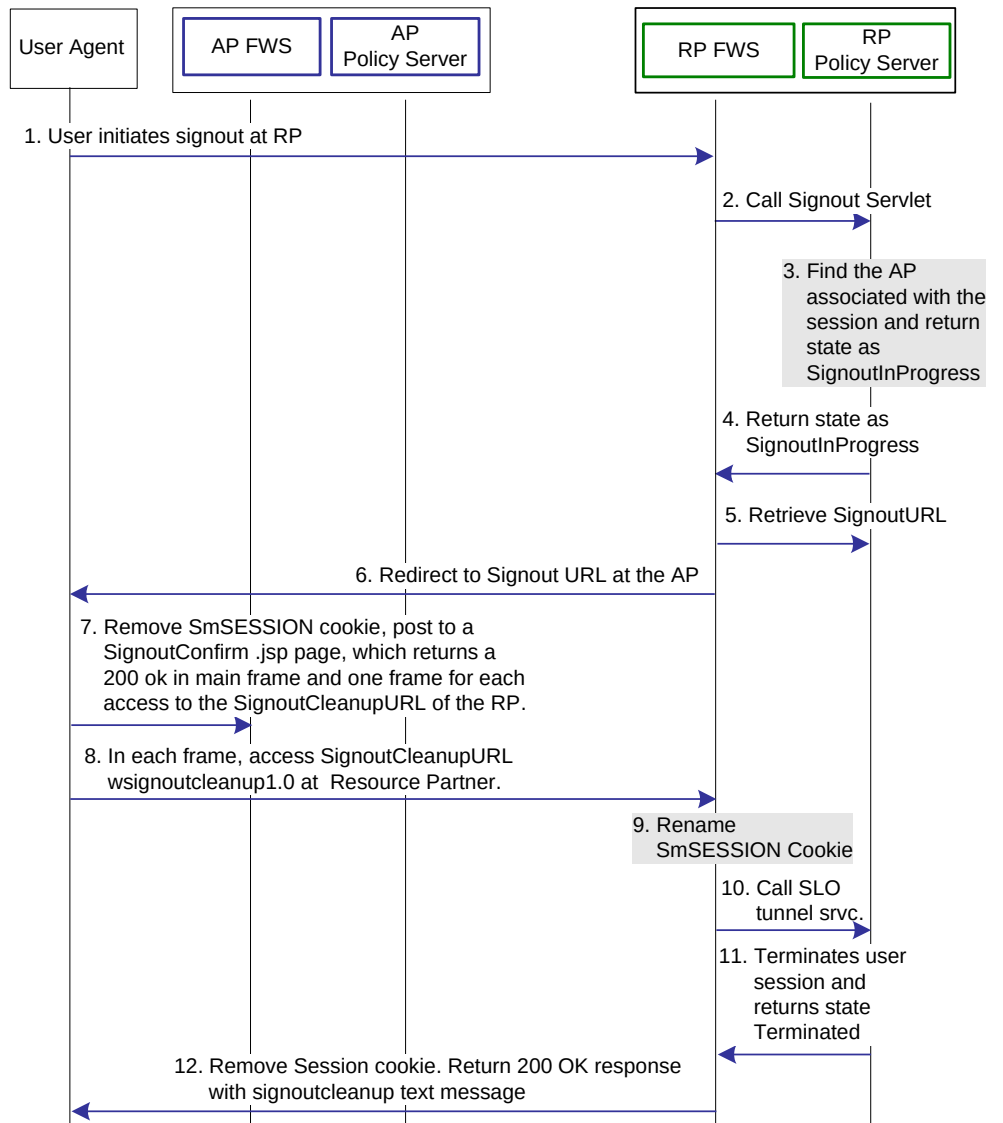
1. The user clicks on a link at the Account Partner to end his global session. The user's browser sends a HTTP-based wsignout request to the signout servlet at the Account Partner.
2. FWS renames the SMSESSION cookie to SESSIONSIGNOUT to invalidate the user's current session.
3. FWS reads the SessionId 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 and 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 RP's provider configuration data, which includes the signout cleanup URL, from the provider's cache 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 main 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 user's browser accesses SignoutCleanup service at the Resource Partner site in an individual frame.
9. When FWS (Signout Servlet) at the Resource Partner receives a wsignoutcleanup request, it renames the SMSESSION cookie to SESSIONSIGNOUT and 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 illustration that follows shows the detailed flow for a signout request between a user's browser and the Federation Security Service components deployed at an Account Partner (AP) and Resource Partner sites. This set-up enables signout for all entities that have a session with a particular user.

The following diagram 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 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 on a link at the Resource Partner to end his global session. The user's browser sends a HTTP-based wsignout request to the Signout servlet at the Resource Partner.

Note: that 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 the information found in session store, the tunnel library determines that the user session was created by consuming a SAML assertion from an Account Partner. 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 user's browser 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 main 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 user's browser accesses SignoutCleanup service at the Resource Partner site in an individual frame.
9. When FWS (Signout Servlet) at the Resource Partner receives a wsignoutcleanup request, it renames the SMSESSION cookie to SESSIONSIGNOUT and 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.

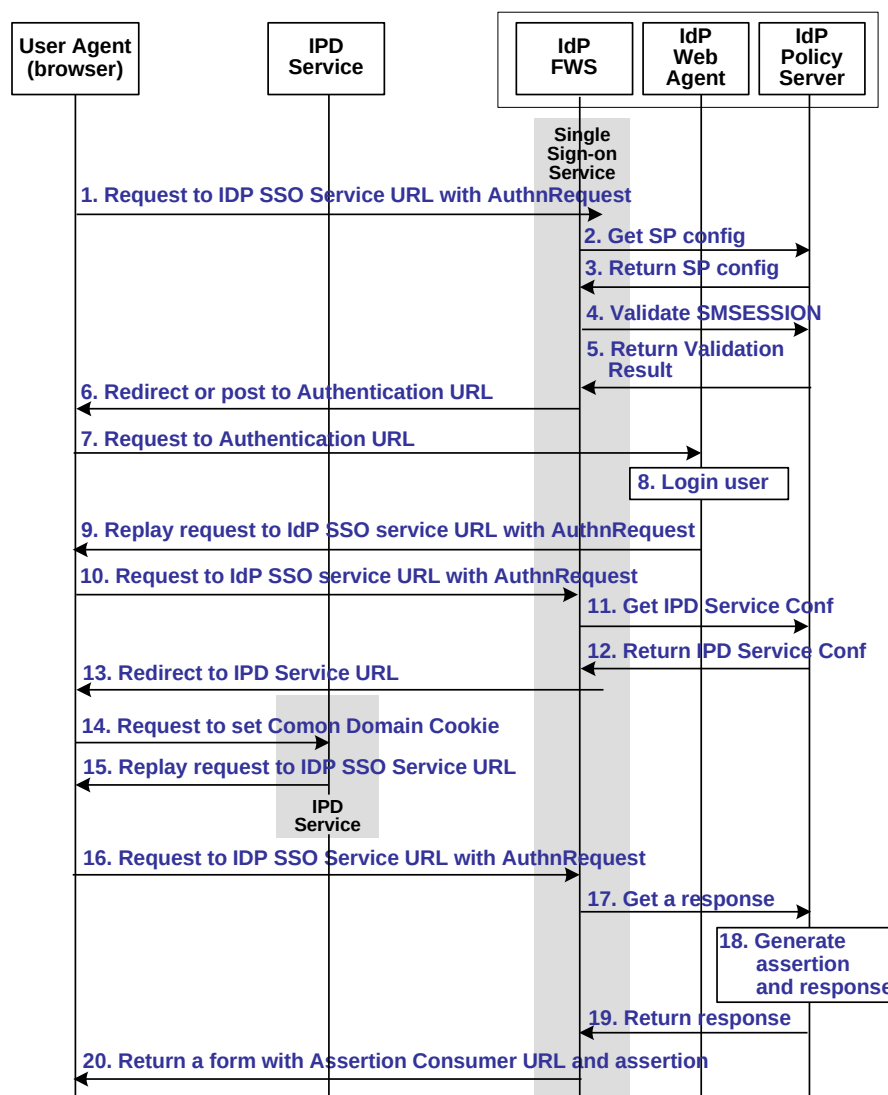
11. Then SLO tunnel library returns FWS with a Terminated status message indicating that the user session no longer exists in the session store.
12. The FWS Signout Servlet removes the SESSIONSIGNOUT cookie and returns a 200 OK response in the frame.

**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 illustration that follows shows the detailed flow for an Identity Provider Discovery service between a user's browser and the Federation Security Service components deployed at an Identity Provider site. This set-up involves redirecting from an Identity Provider to the Identity Provider Discovery Profile service to set the common domain cookie.

The following diagram 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 SiteMinder 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 *CA SiteMinder 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 that the FWS may 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 information. If the SMSESSION cookie is valid, the IdP FWS skips to Step 18.
7. The user agent requests the Authentication URL, which is protected by the IdP Web Agent.
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/updates the common domain cookie with the Identity Provider's ID and 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 via an authorize call to the realm obtained from the configuration information.
18. The Policy Server generates an assertion based on the configuration information for the Service Provider, signs it, 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 information and Javascript to submit the form.

**Note:** If the assertion generator indicates that the current session's authentication level is too low, the IdP FWS will redirect 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.

## Where to Find Federation Security Services Information

For the following tasks, see the following guides:

| Task                                                                    | SiteMinder Guide                                                                                                          |
|-------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------|
| Installing the Policy Server Option Pack and Web Agent Option Pack      | <i>Policy Server, Web Agent Option Pack Guide</i>                                                                         |
| Configuring Federation Security Services features and components        | <i>Federation Security Services Guide</i>                                                                                 |
| Installing and configuring the SAML Affiliate Agent                     | <i>SAML Affiliate Agent Guide</i>                                                                                         |
| Installing and configuring the SPS Federation Gateway                   | <i>Secure Proxy Server Administration Guide</i>                                                                           |
| Using the Policy Server Management API for Federation Security Services | <i>Scripting Guide for Perl</i><br><i>Developer's Guide for C</i>                                                         |
| Looking up SAML 2.0 metadata                                            | <i>Developer's Guide for C</i><br><i>Online Policy Management API Reference</i><br>(PolicyMgtAPI.pod or PolicyMgtAPI.htm) |
| Looking up WS-Federation metadata                                       | <i>Developer's Guide for C</i>                                                                                            |





# Chapter 2: Deploying Federation with the FSS Sample Application

---

This section contains the following topics:

[Federation Sample Application Overview](#) (see page 113)  
[Prerequisites for Using the FSS Sample Application](#) (see page 115)  
[Sample FSS Network](#) (see page 117)  
[How To Run the Sample Application](#) (see page 118)  
[Modify the FederationSample.conf File](#) (see page 118)  
[SetupFederationSample.pl Command Options](#) (see page 119)  
[Deploy the Sample Application on One System](#) (see page 121)  
[Deploy the Sample Application on Two Systems](#) (see page 122)  
[Test Single Sign-on with the FSS Sample Application](#) (see page 123)  
[Test Single Logout with the FSS Sample Application](#) (see page 124)  
[Review Application-Generated SiteMinder Objects](#) (see page 125)

## Federation Sample Application Overview

The easiest way to become familiar with SiteMinder Federation Security Services is to deploy the SiteMinder FSS sample application and use it to test SAML 2.0 single sign-on and single logout. After running the sample application, you can look at the SiteMinder policy objects created by the sample application and examine the SiteMinder logs containing assertions. Finally, you can use the sample application objects as a basis for configuring your own federation environment.

**Note:** The FSS sample application cannot be used with SAML 1.x.

The federation sample application automates all the configuration tasks you would perform manually to accomplish SAML 2.0 single sign-on and single logout.

In a deployment that includes only SiteMinder Federation Security Services, we recommend that you install all components on a single system acting as an Identity Provider (IdP) and Service Provider (SP). However, the sample application can be installed on two separate machines, one acting as the IdP and the other as the SP.

The sample application contains the following components:

- Configuration files for creating SiteMinder policy objects and SMFE objects

- FederationSample.conf

FederationSample.conf contains configuration settings that define the IdP and SP-side SiteMinder policy objects. The information in this file is also used to create sample web pages to test single sign-on and single logout using the local environment settings.

- SMFEConfig.conf

**Important!** For FSS-to-FSS communication, the SMFEConfig.conf file is not used, but it is installed with the Policy Server and Policy Server Option Pack.

SMFEConfig.conf contains information used to create SMFE objects, such as the SP and IdP connection settings for SMFE-to-FSS communication. The information in this file is also used to create sample web pages to test single sign-on and single logout using the local environment settings.

For information about the SMFEConfig.conf file, see the *SiteMinder Federation Endpoint Deployment Guide*.

- SetupFederationSample.pl Perl script

SetupFederationSample.pl is a Perl script that executes the FSS sample application. This script creates the objects needed for the IdP and SP sites. The script also creates the necessary web pages required 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.

**Note:** By default, the script assumes an FSS-to-FSS configuration.

- Web pages to Test Single Sign-on and Single Logout

The sample application installs web pages with HTML links to trigger SAML 2.0 single sign-on and single logout transactions between the IdP and SP. When you install the sample application, the directories with these pages are copied to the web server's document root directory that you specify in the FederationSample.conf file.

The IdP web pages are in the idpsample directory within the web server's document root. 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.

The SP web pages are in the spsample directory under the web server's document root. 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 with the user's credentials at the idp.demo partner site. 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, located in /spsample/protected directory. It is protected by the SAML 2.0 authentication scheme. A user sees this page when single sign-on between the IdP and SP is successful.

- SLOConfirm.jsp

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

## Prerequisites for Using the FSS Sample Application

Before you run the FSS sample application, you must satisfy the following requirements:

- SiteMinder requirements (SiteMinder r6.0 SP6 components recommended):

- Install an LDAP or ODBC user directory and policy store
- Install a SiteMinder Policy Server
- Install a SiteMinder Web Agent

The web server where the Web Agent is installed does not require an SSL port.

- Ensure the Policy Server and Web Agent are configured properly and that you can protect a resource using any authentication scheme.

**Important!** Core SiteMinder has to be functioning properly to run the sample application successfully.

- SiteMinder Federation Security Services requirements
  - Install the SiteMinder Policy Server Option Pack on the same server as the Policy Server.

Be sure to import the `ampolicy.smdif` file when installing the Policy Server Option Pack.
  - Install the SiteMinder Web Agent Option Pack on the same web server as the Web Agent. This is required because Federation Web Services, which is installed by the Web Agent Option Pack, has to run on the same system as ServletExec, the servlet engine being used with the sample application.
  - (Optional) If you want to see the assertion that is generated when you test single sign-on, you have to enable FSS trace logging so that the `FWSTrace.log` file is generated. You enable logging in the `LoggerConfig.properties` file, located in the directory `web_agent_home/affwebservices/WEB-INF/classes`.
  - Install and configure ServletExec.

You should have received a license file called `licensekey50.txt` from CA Support. From this license file, copy the license key and enter it in the ServletExec License dialog of the ServletExec Administration Console.
  - Deploy Federation Web Services as a ServletExec web application.

You do not have to go through the setup steps for the Federation Web Services Application described in the *Federation Security Services Guide*; these tasks are done by the sample application. However, you do have to deploy Federation Web Services as a ServletExec web application. You deploy a web application using the ServletExec Administration Console.

**Note:** For instructions on installing ServletExec, setting the license key, and deploying web applications, see New Atlanta Communication's ServletExec documentation.
  - Configure and enable a session store (SQL or Oracle ODBC database). Setup the database instance using a session server scheme file.

For instructions on configuring a session server, see the *SiteMinder Policy Server Installation Guide*. To point the Policy Server to the session server after you have created it, see the *SiteMinder Policy Server Management Guide*.

**Note:** If you are using two separate systems, install all these components on both systems.

- Use the Perl interpreter shipped with the FSS sample application to run the application.
- Add the variable `<NETE_PS_ROOT>\CLI\bin` to the system PATH environment variable to invoke the Perl script to run the sample application.

**Important!** Ensure that the Perl binary bundled with the SiteMinder Policy Server Option Pack is the first or only such binary in the PATH so that the bundled Perl binary is invoked and not another Perl script.

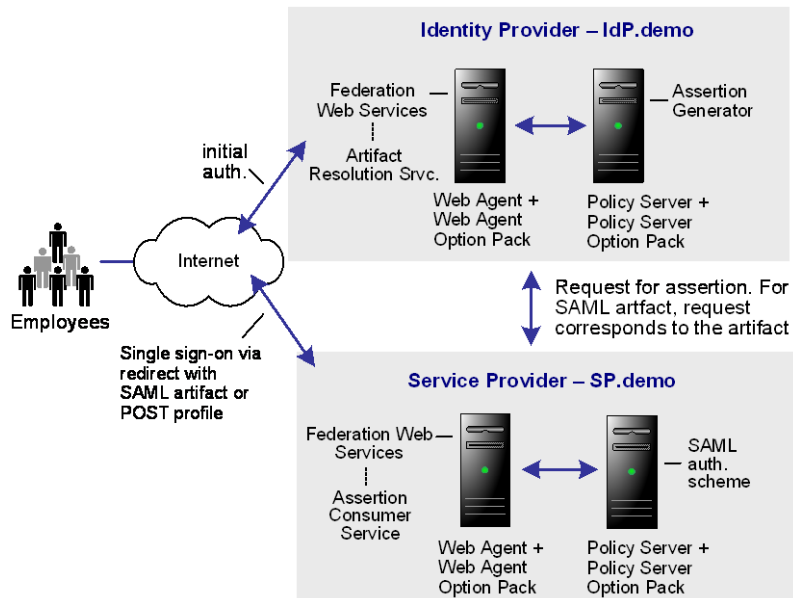
- The user running the Perl script must have read/write permissions to the web server's document root directory.

**Note:** If your Federation Security Services system is communicating with a SiteMinder Federation Endpoint system, see the *CA SiteMinder Federation Security Services Federation Endpoint Guide* for the additional Federation Endpoint requirements.

## Sample FSS Network

The sample Web sites in the SiteMinder federated network are an Identity Provider named `idp.demo`, and a Service Provider named `sp.demo`. There is a business partnership between `idp.demo` and `sp.demo`.

The following illustration shows the sample federated network.



## How To Run the Sample Application

After completing the necessary prerequisites, you can run the sample application.

**To use the FSS sample application, complete the following process**

1. Modify the settings in the FederationSample.conf file for your environment.
2. Run the SetupFederationSample.pl Perl script.
3. Test single sign-on and single logout.

## 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. It also contains one setting for the partner site.

At an FSS site, the information in this file is used to create IdP and SP SiteMinder policy objects, such as policy domains and SAML Service Provider objects.

**To access the FederationSample.conf file and modify the settings**

1. Go to *policy\_server\_home/samples/federation*.
2. Open the FederationSample.conf file.
3. Modify the settings in the file.
4. Save the file.

## FederationSample.conf Settings

You must configure all the settings in the FederationSample.conf file at an FSS site.

The settings are as follows:

### **USER\_DIRECTORY**

Specifies the name of an existing user directory object specified in the Policy Server User Interface. 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 there is only one user directory 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. There is no default value.

**USER\_ATTRIBUTE**

Indicates that the value of this attribute becomes the Name Identifier 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 web server's document root directory. 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 Web server's listening port. 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 is the SP's web server port. The default port is 80.

## SetupFederationSample.pl Command Options

The SetupFederationSample.pl script executes the sample application and sets up the IdP and SP objects that enable single sign-on and single logout.

This script is located in the directory *policy\_server\_home/samples/federation*

To run the sample application script, use the following command and the associated command options.

The command syntax is:

```
perl SetupFederationSample.pl -command_option value
```

You can specify several command options in a command line.

**Example:**

```
perl SetupFederationSample.pl -idp FSS
```

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

The SetupFederationSample.pl command options are:

**-admin**

Specifies the SiteMinder Administrator's user name. Use this option only when you are setting up a SiteMinder FSS system.

**-password**

Specifies the SiteMinder Administrator's password in clear text. Use this option only when you are setting up a SiteMinder FSS system.

**-remove**

Removes all objects created by the sample application.

**-idp**

Creates only the IdP objects in the SiteMinder 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 FSS-to-FSS communication. The possible values are FSS or SMFE.

**-sp**

Creates only SP policy objects in the SiteMinder policy store. You cannot use this option and the -idp option together. The possible values are FSS or SMFE.

**-partner**

(optional) Indicates which application is installed at the partner site. The default is FSS. Possible values are: FSS or SMFE.



## Deploy the Sample Application on One System

To run the FSS sample application, you execute the `SetupFederationSample.pl` script from a command line. Be aware that after you run the `SetupFederationSample.pl` once, running it again deletes the sample policy objects created by the previous execution of the script.

**Important!** You must use the Perl interpreter that is shipped with SiteMinder. This script is located in the directory *policy\_server\_home/CLI/bin*.

### To run the sample application on a single system

1. Complete all core SiteMinder and Federation prerequisites.
2. (Optional) If you are using a web browser on a system that does not have the correct host mappings for `www.idp.demo` and `www.sp.demo`, add these mappings to the system's hosts file.
  - On Windows, the host file is typically located in `WINDOWS\system32\drivers\etc\hosts`.
  - On Solaris/UNIX, the host file is commonly located in `/etc/hosts`.
3. Define your environment by configuring the [FederationSample.conf file](#) (see page 118).
4. Open up a command window.
5. Navigate to *policy\_server\_home/siteminder/samples/federation*
6. Enter the following command then follow the prompts:

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

**Note:** When you are prompted to continue with the installation, enter the word "yes." Do not only enter the letter "y."
7. Restart the Policy Server after the script is finished.
8. Test single sign-on and [single logout](#) (see page 124).

The script accomplishes the following:

- reads the configuration information from `FederationSample.conf` file.
- creates policy objects in the policy store that are needed 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 and the corresponding certificate data to `smkeydatabase`.
- modifies the system's hosts file to map a loopback IP address, `127.0.0.1` to `www.sp.demo` and `www.idp.demo`

## Deploy the Sample Application on Two Systems

You can install the sample application on two separate systems, one system acting as the Identity Provider and the other system acting as the Service Provider. Be aware that after you run the `SetupFederationSample.pl` once, running it again deletes the sample policy objects created by the previous execution of the script.

### To execute the sample application on two systems

1. Complete all the core SiteMinder and Federation prerequisites on both systems.
2. Modify the host file of each system so it recognizes the other system with which it is communicating.
  - On the IdP system, `www.idp.demo`, modify the host file of this system to include the IP address of the SP system.
  - On the SP system, `www.sp.demo`, modify the host file of this system to include the IP address of the IdP system.

On Windows, the host file is typically located in `WINDOWS\system32\drivers\etc\hosts`.

On Solaris/UNIX, the host file is commonly located in `/etc/hosts`.

3. (Optional) If you are using a web browser on a system that does not have the correct host mappings for `www.idp.demo` and `www.sp.demo`, add these mappings to the system's hosts file.
4. Define your environment by configuring the [FederationSample.conf file](#) (see page 118) for each system.
5. Execute the sample application as follows:
  - On the IdP system, enter the following command:

```
perl SetupFederationSample.pl -admin siteminder_administrator  
-password administrator_password -idp FSS
```
  - On the SP system, enter the following command:

```
perl SetupFederationSample.pl -admin siteminder_administrator  
-password administrator_password -sp FSS
```
6. Restart both Policy Servers.
7. Test single sign-on and [single logout](#) (see page 124).

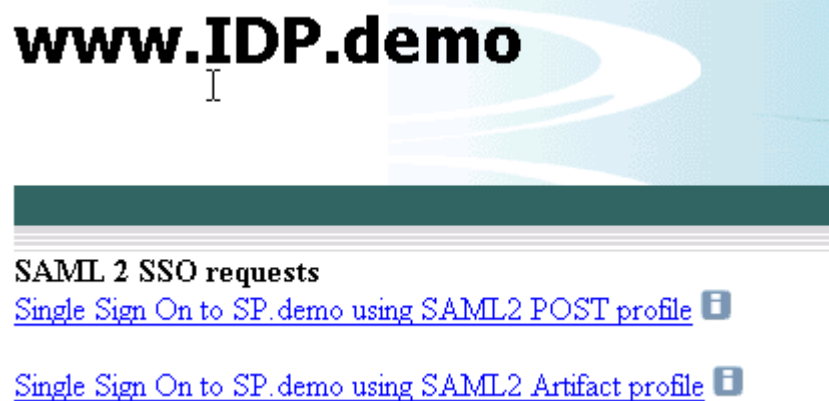
## Test Single Sign-on with the FSS Sample Application

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

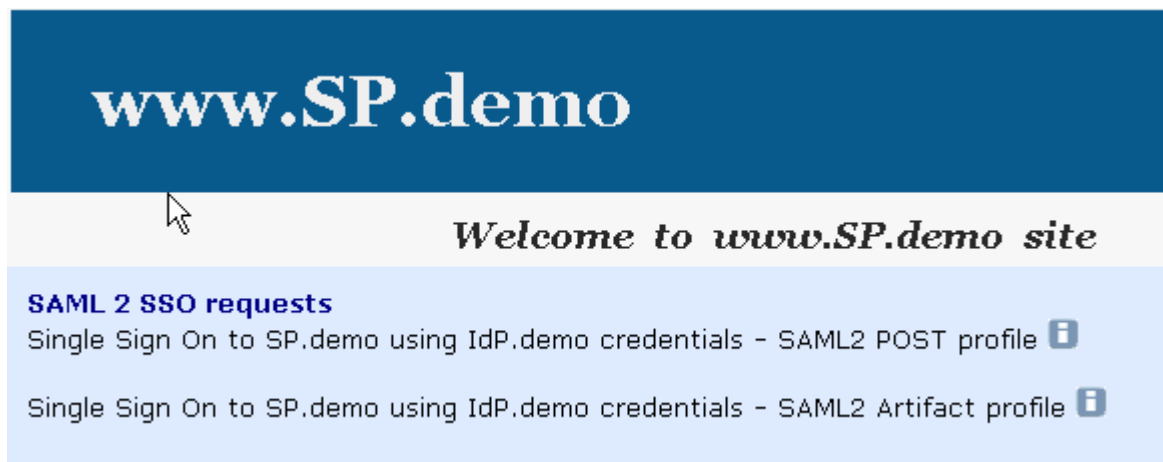
### To test federated single sign-on

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:

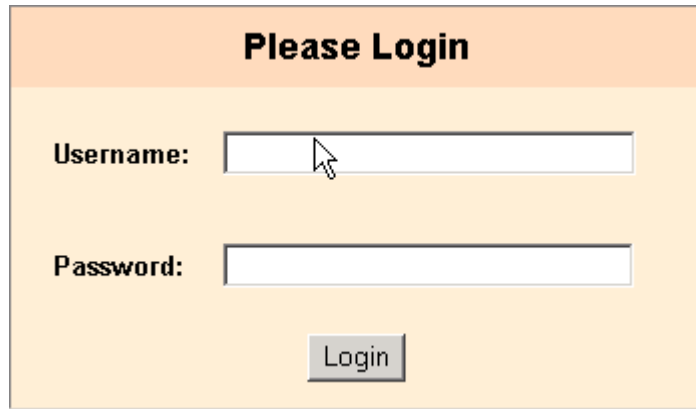


The following figure is the SP.demo home page:



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

A login challenge like the following is presented:

A login form with an orange header bar containing the text "Please Login". Below the header, there are two input fields: "Username:" and "Password:". A mouse cursor is pointing at the "Username:" field. Below the input fields is a "Login" button.

**Please Login**

Username:

Password:

Login

4. Using the login of an existing user in your user store, enter the user's credentials.  
For example, if Tuser1 is a user in the user store, enter the credentials for this user.

If single sign-on is successful, you should see the following welcome page:



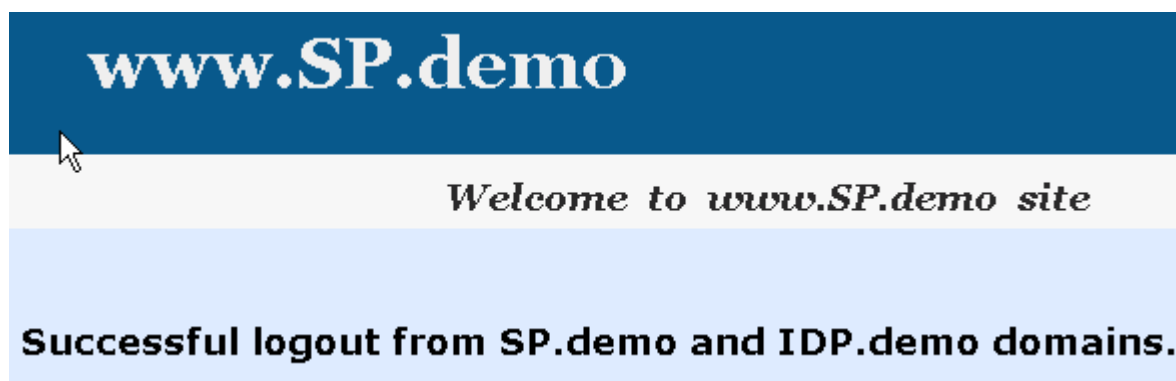
## Test Single Logout with the FSS Sample Application

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 on the link labeled Single Logout using HTTP Redirect binding.

The following page is displayed:



If you see this message, single logout is successful.

## Review Application-Generated SiteMinder Objects

The FSS sample application automatically creates policy objects that enable the federated single sign-on and logout processes. After successfully signing on, log on to the Policy Server User Interface 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  
This is the Service Provider Properties object created by the sample application
- Partner Idp.demo Auth Scheme  
This is the SAML Authentication Scheme created for the SP-site by the sample application.

To see the SAML assertion generated by SiteMinder, look at the FWSTrace.log, located in the directory *web\_agent\_home/log*.

**Note:** You have to 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: Deploying Federation without the FSS Sample Application

---

This section contains the following topics:

[Manual FSS-to-FSS Deployment Overview](#) (see page 127)

[Manual Deployment Prerequisites](#) (see page 128)

[Sample Federation Network](#) (see page 128)

[Set Up the Identity Provider](#) (see page 133)

[Set Up the Service Provider](#) (see page 149)

[Test SAML 2.0 Single Sign-on](#) (see page 160)

[Add Functionality to the Federation Deployment](#) (see page 163)

## Manual FSS-to-FSS Deployment Overview

You can accomplish manually what the sample application deploys automatically. 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 artifact binding, digital signing, and encryption are described so 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 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 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 have to know how to use the Sun ONE Server Console.
- Set up an ODBC database to be enabled as a session store.

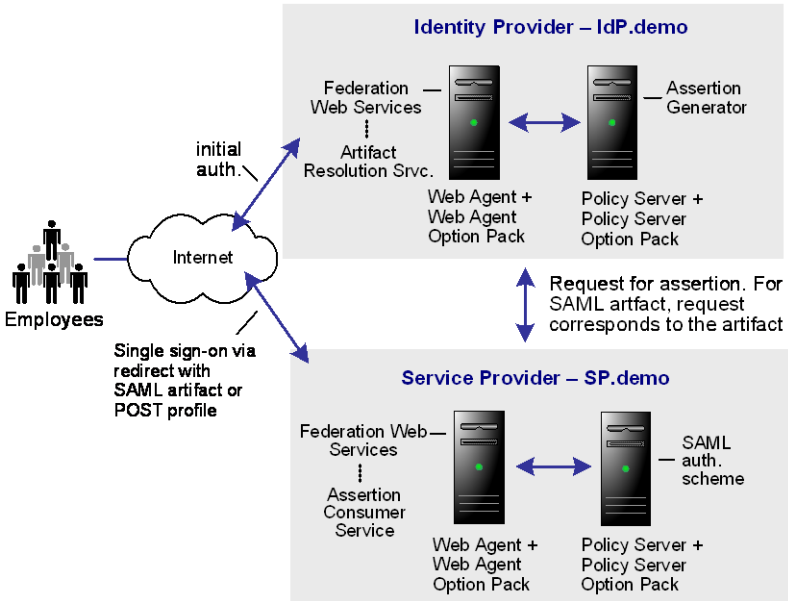
For instructions on setting up a session store database, see the *Policy Server Installation Guide*. Enable the session store using the Policy Server Management Console.

## Sample Federation Network

The sample Web sites in the SiteMinder federated network are an Identity Provider named idp.demo, and a Service Provider named sp.demo. There is a business partnership 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 site’s set-up for a basic SAML 2.0 POST configuration and then a more advanced configuration. You can also fill in information for your network.

The following table contains information 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<br>Server type: IIS 6.0 Web Server | Server:<br>Server type: |

| Identity Provider Component              | Sample Network                                                                                                                                                                                                                                                                                                                                                                                               | Your Network                                                                                                                   |
|------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------|
| IdP policy store                         | <b>IP Address:</b> www.idp.demo:389<br><b>Storage:</b> LDAP<br>(Sun One Directory Server)<br><b>Root DN:</b> o=idp.demo<br><b>Admin Username:</b> cn=Directory Manager<br><b>Password:</b> federation                                                                                                                                                                                                        | IP Address:<br>Storage:<br>Root DN:<br>Admin Username:<br>Password:                                                            |
| User store                               | <b>Server:</b> www.idp.demo:42088<br><b>Server Type:</b> Sun One Directory Server<br>(LDAP)<br><b>User store:</b> The LDAP directory contains the following users: <ul style="list-style-type: none"> <li>■ Tuser1</li> <li>■ Tuser2</li> </ul> <b>userpassword:</b> test<br><b>mail:</b> <user_name>@idp.demo<br><b>Root:</b> dc=idp,dc=demo<br><b>Start:</b> uid=<br><b>End:</b> ,ou=People,dc=idp,dc=demo | Server:<br>Server Type:<br>User store:<br>Users' passwords:<br>Attribute:<br>Attribute description:<br>Root:<br>Start:<br>End: |
| IdP Web Agent with Web Agent Option Pack | <b>Server:</b> www.idp.demo:80<br><b>Server Type:</b> IIS 6.0 Web Server<br><b>Agent name:</b> idp-webagent                                                                                                                                                                                                                                                                                                  | Server:<br>Server Type:<br>Agent name:                                                                                         |
| Assertion Consumer Service URL           | <b>URL:</b><br>http://www.sp.demo:81/affwebservices/public/saml2assertionconsumer                                                                                                                                                                                                                                                                                                                            | <b>URL:</b>                                                                                                                    |
| Assertion Retrieval Service URL          | <b>URL:</b><br>http://www.idp.demo:80/affwebservices/assertionretriever                                                                                                                                                                                                                                                                                                                                      |                                                                                                                                |
| Authentication URL                       | <b>URL:</b><br>http://www.idp.demo/siteminderagent/redirectjsp/redirect.jsp                                                                                                                                                                                                                                                                                                                                  | <b>URL:</b>                                                                                                                    |

## 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 as well as signing and encrypting assertions.

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

## Service Provider Data for a Basic Configuration

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

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

| Service Provider Component             | Sample Network                                                                                                                                                                                                                                                                                                                                                                                               | Your Network                                                                                                                                     |
|----------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------|
| SP Policy Server                       | <b>Server:</b> www.sp.demo:80<br><b>Server type:</b> IIS 6.0 Web Server                                                                                                                                                                                                                                                                                                                                      | Server:<br>Server type:                                                                                                                          |
| SP policy store                        | <b>IP Address:</b> www.sp.demo:389<br><b>Storage:</b> LDAP (Sun One Directory Server)<br><b>Root DN:</b> o=ca.com<br><b>Admin Username:</b> cn=Directory Manager<br><b>Password:</b> federation                                                                                                                                                                                                              | IP Address:<br>Storage:<br>Root DN:<br>Admin Username:<br>Password:                                                                              |
| User Store                             | <b>Server:</b> www.sp.demo:32941<br><b>Server Type:</b> LDAP (Sun One Directory Server)<br><b>User store:</b> The LDAP directory contains the following users:<br><ul style="list-style-type: none"> <li>■ Tuser1</li> <li>■ Tuser2</li> </ul> <b>userpassword:</b> customer<br><b>mail:</b> <user_name>@sp.demo<br><b>Root:</b> dc=sp,dc=demo<br><b>Start:</b> uid=<br><b>End:</b> ,ou=People,dc=sp,dc=demo | Server:<br>Server Type:<br>User store:<br>User passwords:<br>Users' password:<br>Attribute:<br>Attribute description:<br>Root:<br>Start:<br>End: |
| SP Web Agent and Web Agent Option Pack | <b>Server:</b> www.sp.demo:81<br><b>Server type:</b> Sun ONE 6.1 Web Server<br><b>Agent name:</b> sp-webagent                                                                                                                                                                                                                                                                                                | Server:<br>Server type:<br>Agent name:                                                                                                           |
| Single Sign-on Service                 | <b>SSO Service:</b><br>http://www.idp.demo:80/affwebservices/public/saml2sso                                                                                                                                                                                                                                                                                                                                 | SSO Service:                                                                                                                                     |

| Service Provider Component | Sample Network                                                                     | Your Network |
|----------------------------|------------------------------------------------------------------------------------|--------------|
| Target Resource            | <b>Target Resource:</b><br>http://www.sp.demo:81/<br>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 as well as signing and encrypting assertions.

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

The first task is to set up the Policy Server at the IdP.

### To install the Policy Server

1. Install a Policy Server.

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

2. Select the Web server for the Policy Server User Interface.

In this deployment, IIS 6.0 Web server is the server on which the Policy Server is installed. Your network may use a different web server.

3. Select a policy store.

In this deployment, an LDAP directory is the policy store.

4. Go to Set up the IdP LDAP Policy Store.

## Point the Policy Server to the IdP LDAP Policy Store

In this deployment, an LDAP policy store is being used.

### To ensure the Policy Server is pointing to the LDAP policy store

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 135).

## Set Up the IdP User Store

At the IdP, you must have a user store with users defined. The assertion generation 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 the tool used to add users to this user store.

**To configure the user store**

1. Add the following users:
  - Tuser1
  - Tuser2
2. Fill-in the attributes for Tuser1 and Tuser2 as follows:

| Tuser1                | Tuser2                |
|-----------------------|-----------------------|
| userpassword: test    | userpassword: test    |
| mail: Tuser1@idp.demo | mail: Tuser2@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 136).

## Install the IdP Policy Server Option Pack

### To install the Policy Server Option Pack

1. Install the Policy Server Option Pack.

**Note:** The Policy Server Option Pack requires a JRE. See the SiteMinder 6.0 Platform Support Matrix on the [Technical Support site](#) for the correct JRE version.

Be sure to import the affiliate data during the installation. To verify that the import is successful, log in to the Policy Server User Interface and click on Domains in the System tab. You should see the FederationWebServices domain object.

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

2. Ensure that the NETE\_JRE\_ROOT environment variable is set and that it points to the JVM.
  - a. Access your system's environment variables.
  - b. Locate the NETE\_JRE\_ROOT variable.
  - c. Set it to the appropriate location of the JVM.
3. (Optional) [Enable Policy Server Trace Logging at the IdP](#) (see page 136) so you can use the log to troubleshoot your setup

## Enable Policy Server Trace Logging at the IdP

At the IdP, enable logging for the Policy Server so you can view the log file smtracedefault.log and examine trace messages about single sign-on and single logout. This log file is located in the directory *policy\_server\_home/siteminder/log*.

### To enable trace logging for federation

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.

You configure trace logging at the Policy Server using the Policy Server Management Console.
3. [Install the IdP Web Agent](#) (see page 137).



## Install the IdP Web Agent

### To install the Web Agent at the IdP

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 6.0 Web server. Your server may be different.

2. Register the machine 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 6.0 Web Server.

4. [Install the IdP Web Agent Option Pack](#) (see page 137).

## Install the IdP 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 the Web Agent Option Pack on the same Web server as the Web Agent. In this deployment, the server is an IIS 6.0 Web Server.

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

2. [Configure the Web Server with the Web Agent Option Pack](#) (see page 137).

## Configure the Web Server with the Web Agent Option Pack

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

### For FWS to work, do the following

- [Install the JDK for Federation Web Services](#) (see page 138)
- Install and Configure ServletExec to work with FWS at the IdP
- [Configure the AffWebServices.properties File at the IdP](#) (see page 140)
- [Test Federation Web Services at the IdP](#) (see page 141)

## 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 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: You should have received a ServletExec license key file called licensekey50.txt from CA 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 web site.

Be sure to apply the most current hot fixes for the supported version of ServletExec you are using. Without the hot fixes, Federation Web Services will not work with ServletExec. To obtain hot fixes, go to the web site for [New Atlanta Communication](#).

### To set up ServletExec

1. Install ServletExec. Refer to New Atlanta's documentation for instructions.
2. Open the ServletExec 5.0 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 may 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!** IIS does not allow any plug-in to write to a file system unless it is configured with a user account that has proper rights to do so. Therefore, for Federation Web Services to work with ServletExec, you need to modify the directory security settings for the IIS default user account.

**More Information:**

[Enable ServletExec to Write to the IIS File System](#) (see page 139)

[Configure the AffWebServices.properties File at the IdP](#) (see page 140)

## Enable ServletExec to Write to the IIS File System

The IIS Web server does not allow a plug-in to write to its file system unless it is configured with a user account that has proper rights to do so. Therefore, for ServletExec to write to the federation log files, the anonymous user account that you associate with ServletExec must have permissions to write to file system.

**To enable the user account used by ServletExec to write to the IIS file system**

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 group box.  
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. Refer to Windows documentation to grant this right to a user account. For example, you might use the IUSR Internet Guest account for anonymous access.
  - b. Deselect Basic authentication.
  - c. Deselect 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 associated with ServletExec can now write to the IIS file system.

Additionally, you must give the anonymous user the right to act as part of the operating system.

To give the anonymous user account the right to act as part of the operating system

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. Although it is optional, we strongly recommend that you, look at the Agent Configuration Object for the Web Agent protecting the IIS Web server and ensure that the SetRemoteUser parameter is set to yes to make sure that the anonymous user can write to the file system.

## Configure the AffWebServices.properties File at the IdP

The affwebservices.properties file contains all the initialization parameters for Federation Web Services. You need to 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 C:\Program Files\ca\webagent\affwebservices\WEB-INF\classes
2. Set the AgentConfigLocation parameter to the location of the WebAgent.conf file. It is mandatory to set a value for this parameter.

For this deployment, an IIS 6.0 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 back-slashes. This applies only to Windows.

Ensure this path is entered on one line.

3. Save and close the file.
4. [Test Federation Web Services at the IdP](#) (see page 141).

## Test Federation Web Services at the IdP

After setting up Federation Web Services, ensure it is operating correctly.

### To test that Federation Web Services is operating

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

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

where *fqhn* is the fully-qualified host name and *port\_number* is 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, you should see a message that reads:

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 will get a message that the Assertion Retrieval Service has failed. In case of failure, check the Federation Web Services log.

2. [Enable Web Agent Option Pack Logging at the IdP](#) (see page 141).

## Enable Web Agent Option Pack Logging at the IdP

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

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

### To enable logging

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](#) (see page 142).

### More Information:

[Set up the LoggerConfig.properties File](#) (see page 215)

[Federation Security Services Trace Logging](#) (see page 471)

## Specify the User Store for the IdP Policy Server

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

The following steps specify how to configure a user directory in the Policy Server User Interface. The directory, called IdP LDAP, is the Sun ONE LDAP directory that contains the users Tuser1 and Tuser2.

### To configure a user directory

1. Log into the Policy Server User Interface.
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 group box:

#### **Name**

IdP LDAP

In the Directory Setup group box:

#### **NameSpace**

LDAP

#### **Server**

www.idp.demo:42088

5. Complete the following field in the LDAP Search group box:

#### **Root**

dc=idp,dc=demo

Accept the defaults for the other values.

Complete the following field in the LDAP User DN Lookup group box:

#### **Start**

uid=

#### **End**

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

6. Click View Contents to ensure you can view the contents of the directory.
7. Click Submit.
8. [Set up an Affiliate Domain at the IdP](#) (see page 143).

## 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 into the Policy Server User Interface.
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 group box, select Affiliate Domain.
5. Leave this dialog open and [Add the User Directory to the Affiliate Domain at the IdP](#) (see page 143).

## Add the User Directory to the Affiliate Domain at the IdP

### To associate a user directory for the affiliate domain

1. Begin at the User Directories tab in the Domain Properties dialog.
2. From the drop-down list box at the bottom of the dialog, select IdP LDAP and click Add.

For your network, select the user store you set up at the IdP.

3. Click OK.
4. Go to [Add the Service Provider to the Affiliate Domain at the IdP](#) (see page 143).

## Add the Service Provider to the Affiliate Domain at the IdP

To add sp.demo to the affiliate domain, you must 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. This URL must also be protected by a SiteMinder policy.

**Enabled**

Ensure it is checked. It should be checked by default.

4. Keep the Policy Server User Interface open and [Select Users For Which Assertions Will Be Generated at the IdP](#) (see page 144).

## Select Users For Which Assertions Will Be Generated at the IdP

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 will generate SAML assertions. You may only add users and groups from directories that are in an affiliate domain.

**To select users that will use assertions as credentials**

1. Log in to the Policy Server User Interface.
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 are the employees 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. Holding the CTRL or SHIFT key, 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 145).

## 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.

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 group box**

User Attribute

#### **Attribute Name**

mail

2. Keep the SAML Service Provider Properties dialog open and [Identify the SP, IdP, and Other General Settings](#) (see page 146).

## Identify the SP, IdP, and Other General Settings

You must 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, check 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; you must enable it.

3. Keep the SAML Service Provider Properties dialog open and [Configure POST Single Sign-on at the IdP](#) (see page 146).

### More Information:

[Configure Digital Signing \(required for POST Binding\)](#) (see page 174)

## 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 147).

## 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.

**More Information:**

[Protect the Authentication URL to Create a SiteMinder Session \(SAML 1.x\)](#) (see page 265)

## Federation Web Services Access

The Web Agent that protects the Federation Web Services application needs to be bound to the realms associated with Federation Web Services, which are created by the Web Agent Option Pack installation. To associate the Web Agent with the realms, the Agent needs to be added to the default Web Agent group also created by 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 149).

## Set Up the Service Provider

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

## Install the SP Policy Server

At the Service Provider, you need to install 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 for the Policy Server User Interface.

In this deployment, the Web server is an IIS 6.0 Web server. Your network may use a different server.

3. Select a policy store.

In this deployment, it is an LDAP policy store.

## Point the Policy Server to the SP LDAP Policy Store

### To set up the LDAP policy store

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 151).

## Set Up the SP User Store

At the SP, configure a user store and add user records for users that require assertions. When the user's 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 and the Sun ONE Server Console is the tool used to add users to the directory.

### To configure the user store

1. Add the following users:
  - Tuser1
  - Tuser2
2. Fill-in the attributes for Tuser1 and Tuser2 as follows:

| Tuser1                 | Tuser2                 |
|------------------------|------------------------|
| userpassword: customer | userpassword: customer |
| mail: Tuser1@sp.demo   | mail: Tuser2@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 152).

## Install the SP Policy Server Option Pack

After installing the Policy Server, you need to install the Policy Server Option Pack.

### To install the Policy Server Option Pack

1. Install the Policy Server Option Pack.

**Note:** The Policy Server Option Pack requires a JRE. See the SiteMinder 6.0 Platform Support Matrix on the [Technical Support site](#) for the correct JRE version.

Be sure to import the `ampolicy.smdif` file during the installation; this sets up affiliate policy store objects. To verify that the import is successful, log in to the Policy Server User Interface and click on Domains in the System tab. You should see the `FederationWebServices` domain object.

For instructions on installing the Policy Server Option Pack, see the *Policy Server, Web Agent Option Pack Release Notes*.

**Important!** In this example, import the `ampolicy.smdif` file if you are setting up a new federation configuration. If you have an existing configuration, importing the `ampolicy.smdif` file will reset all affiliate data back to the default settings.

2. Ensure that the `NETE_JRE_ROOT` environment variable is set and that it points to the JVM.
  - a. Access your system's environment variables.
  - b. Locate the `NETE_JRE_ROOT` variable.
  - c. Ensure that the installer has set it to the appropriate location of the JVM.
3. (Optional) [Enable Policy Server Trace Logging at the SP](#) (see page 152) so you can use the log to troubleshoot your setup.

## 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 153).



## Install the SP Web Agent

### To install the Web Agent at the Service Provider

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 may be different.

2. Register the machine 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](#) (see page 153).

## 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 6.0 Platform Support Matrix on the [Technical Support site](#).

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

In this deployment, the server is an IIS 6.0 Web Server.

For instructions on installing the Web Agent Option Pack, see the *Policy Server, Web Agent Option Pack Release Notes*.

2. [Configure the Web Server with the Web Agent Option Pack](#) (see page 153).

## Configure the Web Server with the Web Agent Option Pack

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

### For FWS to work, do the following

1. [Install the JDK for Federation Web Services](#) (see page 154)
2. Install and Configure ServletExec to Work with FWS at the SP
3. [Configure the AffWebServices.properties file](#) (see page 155)
4. [Enable Web Agent Option Pack logging](#) (see page 155)
5. [Test Federation Web Services](#) (see page 156)

## 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, install ServletExec, JBOSS, WebLogic Application Server, or WebSphere Application Server. For this deployment, ServletExec is installed on a Sun ONE 6.1 web server.

Note: You should have received a ServletExec license key file called licensekey50.txt from CA 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 Communications <http://www.newatlanta.com> web site.

Apply the most current hot fixes for the supported version of ServletExec. Without the hot fixes, Federation Web Services will not work with ServletExec. To obtain the hot fixes, go to the web site 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 5.0 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 network may be different. Enter the correct location.

6. Click Submit.

7. Exit the ServletExec Console.
8. [Configure the AffWebServices.properties file](#) (see page 155).

## Configure the AffWebServices.properties file

The AffWebServices.properties file contains all the initialization parameters for Federation Web Services.

### To set up the AffWebServices.properties file

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

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 back-slashes. This entry should be on one line.

3. Save and close the file.
4. [Test Federation Web Services](#) (see page 156).

## 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:**

[Set up the LoggerConfig.properties File](#) (see page 215)

[Federation Security Services Trace Logging](#) (see page 471)

## Test Federation Web Services

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

**To test that Federation Web Services is operating**

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

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

where *fqhn* is the fully-qualified host name and *port\_number* is 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, you should see a message that reads:

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 will get a message that the Assertion Retrieval Service has failed. In case of failure, check the Federation Web Services log.

2. [Enable Web Agent Option Pack logging.](#) (see page 155)

## Specify the POST Binding Authentication at the SP

For the authentication scheme, you must 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 via 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. The target must be protected by a SiteMinder policy that uses the SAML 2.0 authentication scheme.

3. Check the HTTP-POST check box.
4. Deselect the Enforce Single Use Policy check box.  
Unchecking this box makes the sample network non-compliant with SAML 2.0. If you want to enable the use of the single use policy feature you must 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 158).

## 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 into the Policy Server User Interface.
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 group box:

**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 those 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 175)

## Protect the Target Resource at the SP

After configuring 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 Policy Server User Interface, create a policy domain called Domain for IdP.demo Visitors.
2. Define a Web Agent. In this deployment, the Agent is sp-webagent. This is the Agent protecting 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**

This is 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

The target resource is now protected by SiteMinder.

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 an FSS-to-FSS network, you can use the web pages included with the FSS sample application, provided you have previously run the FSS sample application script. If you do not run the FSS sample application, you have to 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 your web server's document root directory.

If you choose to 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 user's authentication context. 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 by the SAML authentication scheme at the SP. The IdP ID field is located on the Scheme Setup tab of the Authentication Scheme Properties dialog.

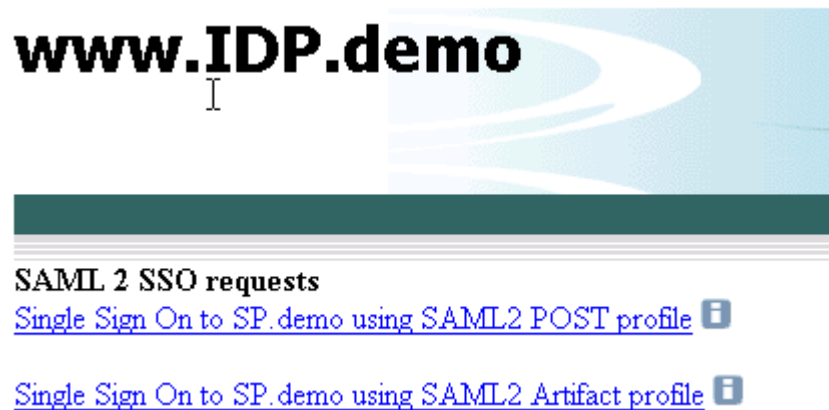


After running the sample application, you should can test single sign-on.

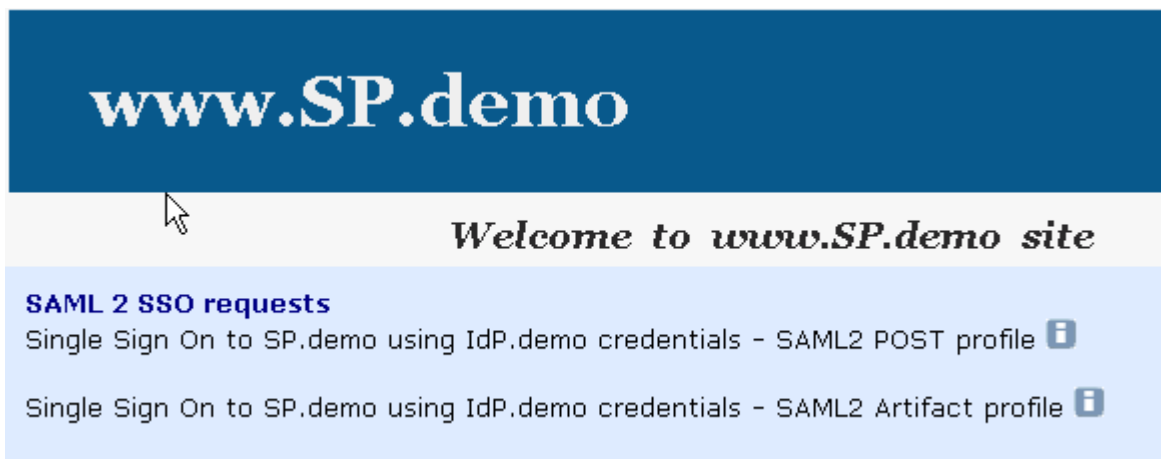
**To test federated single sign-on**

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:

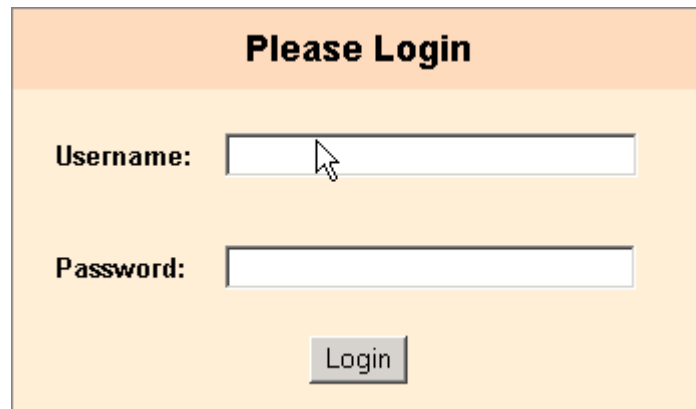


The following figure is the SP.demo home page:



- Click on the SAML2 POST profile link.

A login challenge like the following is presented:

A login form with an orange header bar containing the text "Please Login". Below the header, there are two input fields: "Username:" and "Password:". A mouse cursor is pointing at the Username field. Below the password field is a "Login" button.

**Please Login**

Username:

Password:

Login

- Using the login of an existing user in your user store, enter the user's credentials.  
For example, if Tuser1 is a user in the user store, enter the credentials for this user.

If single sign-on is successful, you should see the following welcome page:



- After you test single sign-on, you can [Add Functionality to the Federation Deployment](#) (see page 163).

## 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
- Configure digital signing of an assertion
- Encrypting and decrypting an assertion

**Note:** Some of these additional features described 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, thereby ensuring security. These sessions must be associated with the browser that initiated the logout. Single logout does not necessarily end all sessions for a user.

**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.

### 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 Policy Server User Interface and access the SAML Service Provider Properties dialog for `sp.demo`.
2. Select the SLO tab.
3. Select the HTTP-Redirect checkbox.

The remaining fields become active.

4. Enter values for the following fields:

**SLO Location URL**

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

This is 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. Ensure 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

You can use the web pages included with the sample application to test single logout, provided you have run the FSS sample application. These 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 your web server's document root directory.

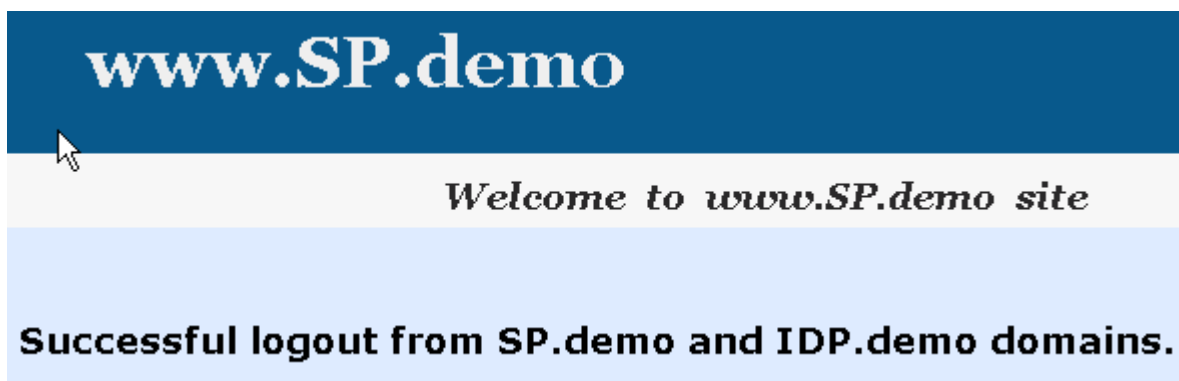
If you have not run the FSS sample application, use your own web pages, but ensure that your HTML page to test SP-initiated single sign-on includes a hard-coded link to the Single Logout service.

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 on the link labeled Single Logout using HTTP Redirect binding.

The following page is displayed:



If you see this message, single logout is successful.

## Configure SAML 2.0 Artifact Single Sign-on

You need to 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 Server for Artifact Single Sign-on](#) (see page 166)
- [Enable SSL for the IdP Web Server for Artifact Single Sign-on](#) (see page 167)
- [Enable a Persistent Session to Store Assertions at the IdP](#) (see page 167)
- [Protect Federation Web Services](#) (see page 168)
- [Select the Artifact Binding at the IdP](#) (see page 169)

Required tasks at the Service Provider:

- [Add a CA Certificate to the Smkeydatabase at the SP](#) (see page 170)
- [Enable the Artifact Binding for SAML Authentication at the SP](#) (see page 171)
- [Add a Link at the SP to Initiate Artifact Single Sign-on](#) (see page 172)
- Test Artifact Single Sign-on

## Set Up the IdP Session Server for Artifact Single Sign-on

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

**Note:** An ODBC database must be used as the session store.

### To enable the session server

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. Choose 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 167).

## 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. This ensures that the back channel over which the assertion is passed is secure.

**To enable SSL at the IdP Web server**

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's 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 167).

## Enable a Persistent Session to Store Assertions at the IdP

You need to 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 147). 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 Policy Server User Interface.
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 radio button.
6. Click OK.
7. [Select the Artifact Binding at the IdP](#) (see page 169).

## Protect Federation Web Services at the IdP (required-POST/Artifact)

Protecting the Federation Web Services application ensures that the services that make up the application are secure.

The policies for the Federation Web Services application are created automatically by the installation of the Web Agent Option Pack. However, to enforce protection and specify who can access Federation Web Services, there are a few additional steps.

### To protect the Federation Web Services application at the IdP

1. Log on to the Policy Server User Interface.
2. Select the System tab.
3. From the menu bar, 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 on Add/Remove and the Available Agents and Groups dialog opens.
8. Add idp-webagent, the IdP Web Agent protecting the Federation Web Services 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 Federation Web Services application to retrieve the assertion, 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 box 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, choose the SP Partners domain and move it to the Current Members list, then click Apply.
- f. Click OK to return to the Policy List.

Federation Web Services is now protected.

## Select the Artifact Binding at the IdP

For artifact single sign-on, you need to enable the artifact binding.

### To configure artifact single sign-on

1. Log in to the Policy Server User Interface.
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/affwebservices/public/saml2assertionconsumer`

6. Select the HTTP-Artifact check box.
7. For the Artifact encoding, select URL.

The artifact will be added to a URL-encoded query string.

8. Complete the password fields:

**Password**

smfederation

**Confirm Password**

smfederation

This is the password that sp.demo will use 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 may want to 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 170).

## 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, you must add a certificate to the Service Provider's smkeydatabase.

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. The Artifact Resolution Service needs to be protected and the back channel need to be secure so the Service Provider knows the SSL connection is secured by a trusted authority.

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. Check whether 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 171).

**Enable the Artifact Binding for SAML Authentication at the SP**

At the Service Provider, you must 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 Policy Server User Interface.
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 172)

## Test SP-Initiated Artifact Single Sign-on

To test single sign-on in an FSS-to-FSS network, you can use the web pages included with the FSS sample application, provided you have previously run the FSS sample application script. If you do not run the FSS sample application, you have to 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 your web server's document root directory.

If you choose to 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
```

where *server:port* is the name and port of the server at the SP where the Web Agent Option Pack is installed and *IdP\_ID* is 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 might look like the following:

```
<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 user's authentication context. 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 by the SAML authentication scheme at the SP. The IdP ID field is located on the Scheme Setup tab of the Authentication Scheme Properties dialog.

You should now be ready to test single sign-on. Follow the steps to [test SP-initiated single sign-on](#) (see page 160).

## Include an Attribute in the Assertion

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

For this deployment, an attribute will be added for Tuser1.

### To add the firstname attribute

1. Log in to the Policy Server User Interface.
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 Tuser1's profile.

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. There are configuration tasks at the Identity Provider and Service Provider to 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 174)

Required tasks at the Service Provider:

- [Set Up the Key Database at the SP to Validate Digital Signatures](#) (see page 175)
- [Enable Signature Validation at the SP](#) (see page 176)

## 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 you need to 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 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 Policy Server User Interface.
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 175).

**More Information:**

[Manage the Key Database for Signing and Encryption](#) (see page 479)

## 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, you need to add a public key to Service Provider's smkeydatabase file. When you configure the SAML authentication scheme, you specify the issuer's DN and serial number of the corresponding partner certificate.

**To add the public key to smkeydatabase**

1. Open a command window.
2. Create the smkeydatabase by entering:  

```
smkeytool.bat -createDB -password password
```

This 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 idplcert -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 176).

**More Information:**

[Manage the Key Database for Signing and Encryption](#) (see page 479)

## Enable Signature Validation at the SP

**To validate a digital signature for POST single sign-on**

1. Log into the Policy Server User Interface.
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 box opens.
3. In the Scheme Common Setup group box, uncheck the Disable Signature Processing check box. Unchecking this box 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 Service Provider's smkeydatabase.



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.

There are configuration tasks at the Identity Provider and Service Provider.

Required tasks at the IdP:

- [Add a Public Key to SMkeydatabase at the IdP](#) (see page 177)
- [Enable Encryption in the Policy Server User Interface at the IdP](#) (see page 177)

Required task at the SP:

- [Decrypt an Encrypted Assertion at the SP](#) (see page 178)

### 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 Policy Server User Interface.
2. From the Service Provider Properties dialog, select the Encryption tab.
3. Check the Encrypt Assertion check box.
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's 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 open a command window and enter the command `smkeytool -listCerts` to list the certificates and view the DN to ensure that you enter a matching value.

6. In the Serial Number field, enter the serial number of the public key that resides in the Identity Provider's smkeydatabase. 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 178).

## 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 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 Partnership Setup

---

This section contains the following topics:

[Installation Overview](#) (see page 181)

[Conventions in the Installation Overview Procedures](#) (see page 182)

[Set Up Producing Authority Components](#) (see page 183)

[Set Up Consuming Authority Components](#) (see page 193)

## Installation Overview

This overview outlines the set up of a SiteMinder federated network.

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

A producing authority can be a:

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

A consuming authority can be a:

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

**Note:** You can perform all the installation tasks first then complete the software configuration via the Policy Server User Interface. Either method works.

These procedures refer to the latest SiteMinder 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

***fqhn***

Designates fully-qualified host name

***port\_number***

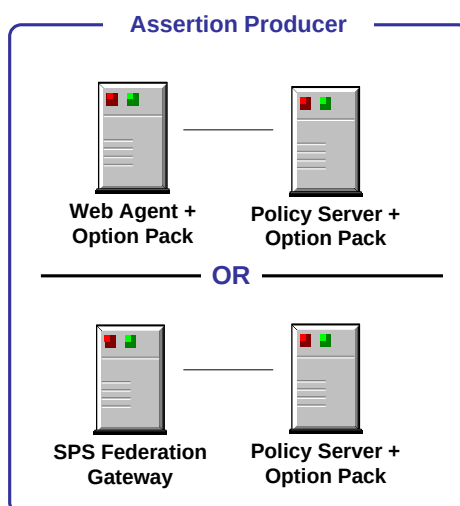
Specifies the port number of a server

***sps\_home***

Specifies the installed location of the Secure Proxy Server

## Set Up Producing Authority 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**  
See *Policy Server Installation Guide*
2. **Install the Policy Server Option Pack**  
See *Policy Server, Web Agent Option Pack Release Notes*
3. **Set up affiliate domains and affiliates/SPs/RPs**  
See *SiteMinder Federation Security Services Guide*
4. **Install and configure a Web Agent or SPS Federation Gateway (skip steps 5 and 6 if using SPS)**  
See *SiteMinder Web Agent Installation Guide* or *SPS Administration Guide*
5. **Install a web or application server for the Web Agent Option Pack**  
See *SiteMinder Federation Security Services Guide*
6. **Install the Web Agent Option Pack**  
See *Policy Server, Web Agent Option Pack Release Notes*
7. **Configure Federation Web Services**  
See *SiteMinder Federation Security Services Guide*
8. **Protect Federation Web Services**  
See *SiteMinder Federation Security Services Guide*
9. **For SAML 2.0 responses, set up a key database for signing**  
See *SiteMinder Federation Security Services Guide*
10. **Create links to target resources at the consumer/SP**  
See *SiteMinder Federation Security Services Guide*

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

## Install the Producing-side Policy Server

At the producing authority:

1. Install the Policy Server.

See *SiteMinder Policy Server Installation Guide*.

2. Set up the Session Server and its database for artifact single sign-on only.

See *SiteMinder Policy Server Management*.

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

**Note:** Do not use Microsoft Access as a session server database.

3. Set up a policy store for use by the Policy Server.

See *SiteMinder Policy Server Installation Guide*.

4. Set up a user store.

See *SiteMinder Policy Design*.

This user store must contain the users for which assertions will be generated.

## Install the Producing-side Policy Server Option Pack

The Policy Server must already be installed prior to installing the Option Pack.

**Note:** The Policy Server Option Pack requires a JDK. For the supported JDK version, log on to the [Technical Support site](#) and search for the SiteMinder 6.0 Platform Support Matrix.

At the producing authority:

1. Install the Policy Server Option Pack.

See the *Policy Server, Web Agent Option Pack Release Notes*.

**Note:** Be sure to import the affiliate data during the installation. The import is successful if the FederationWebServices domain object is in the Domains List in the Policy Server User Interface.

2. Make sure that the NETE\_JRE\_ROOT environment variable is set and that it points to the JVM.

You may want to enable error and trace logging for the Policy Server to see the communication between the producing authority and the consuming authority.



## 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 will consume assertions to the affiliate domains. This identifies the partners to the site producing the assertions.

At the producing authority:

1. Access the Policy Server User Interface.
2. Create an affiliate domain.
3. Add a user store for users that the producing authority (producer, IdP, AP) will generate assertions.
4. Add an object for each consuming authority (consumer, SP, RP) to the affiliate domain.

There should be a one-to-one correspondence between a consuming authority and each object added to the domain.

5. After adding sites to an affiliate domain, ensure that you protect the AuthenticationURL, which ensures that a user has a session at the producing authority prior to process a request for a federated resource.

To do this:

- 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 242)

[Protect the Authentication URL to Create a SiteMinder Session \(SAML 1.x\)](#) (see page 265)

[Protect the Authentication URL to Create a SiteMinder Session \(SAML 2.0\)](#) (see page 343)

## Install a Web Agent or SPS Federation Gateway (Producing-side)

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 producing authority, set up the following components:

1. Install one of the following:
  - Web Agent  
For instructions, see *SiteMinder Web Agent Installation Guide*.
  - SPS federation gateway  
For instructions, see *SiteMinder Secure Proxy Server Administration Guide*.
2. For artifact single sign-on only, SSL-enable the web server with the Web Agent installed or the system with the SPS federation gateway, depending on what is installed in your environment.

If the SAML Affiliate Agent is the consumer, the SSL-enabled web server at the producer must be configured to ignore client certificates. (This is the web server where the Web Agent is installed.) The SAML Affiliate Agent's Affiliate Server component cannot communicate with the Web Agent if the web server is configured to accept client certificates.

## Install a Web or Application Server for the Web Agent Option Pack (Producing-side)

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. To install this component, you need a web or application server.

At the producing authority, do the following:

1. Install one of the following servers to run Federation Web Services, the application 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 Producing-side 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 producing authority, do the following

1. Install the Web Agent Option Pack.

For instructions, see the *Policy Server, Web Agent Option Pack Release Notes*.

2. Be sure to install a JDK, which is required by the Web Agent Option Pack.

For the supported JDK version, log on to the [Technical Support site](#) and search for SiteMinder 6.0 Platform Matrix.

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

## Configure Federation Web Services (Producing-side)

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 producing authority

1. Configure a web server running ServletExec or 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 is already deployed.

2. Check 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 Federation Web Services application. Logging is enabled in the `LoggerConfig.properties` file. The logs enable you to see the communication between the producing authority and the consuming authority.
  - 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`

where *fqhn* is the fully qualified host name and *port\_number* is 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 a message that reads:  
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 the test fails, look at the Federation Web Services log.

**More Information:**

[Configure WebLogic to Work with Federation Web Services](#) (see page 222)  
[Configure WebSphere to Work with Federation Web Services](#) (see page 226)  
[Configure the AffWebServices.properties File](#) (see page 214)  
[Set up the LoggerConfig.properties File](#) (see page 215)  
[Federation Security Services Trace Logging](#) (see page 471)

## Protect Federation Web Services (Producing-side)

You need to protect the Federation Web Services application so that only authorized partners have access.

**To protect the Federation Web Services application**

1. Log in to the Policy Server User Interface at the producing authority.
2. Check the Agent Configuration Object for the Web Agent to ensure the necessary parameters are set.

3. Add the Web Agent that protects Federation Web Services to the Agent group FederationWebServicesAgentGroup. The Web Agent at the producing authority is on the Web server where the Web Agent Option Pack is installed.

This action binds the Agent to the realms that protect Federation Web Services.

Select View, Agent Groups to view Agent groups.

4. Specify the consuming authorities who can access the Federation Web Services application:

- a. From the Domains tab, expand FederationWebServicesDomain and select Policies.

The following policies are displayed in the Policy List:

- FederationWSAssertionRetrieval
- FederationWSNotificationService
- FederationWSSessionServicePolicy
- SAML2FWSArtifactResolutionServicePolicy

- b. Select one of the policies, and click Edit, Properties of Policy.

For SAML 1.x, you need to permit access to:

- FederationWSAssertionRetrieval
- FederationWSNotificationService
- FederationWSSessionServicePolicy

For SAML 2.0, you need to permit access to:

- SAML2FWSArtifactResolutionServicePolicy

**Note:** You do not have to select a policy for WS-Federation.

The SiteMinder Policy dialog box opens.

- c. From the Users tab, select the FederationWSCustomUserStore tab for SAML 1.x or the SAML2FederationCustomUserStore tab for SAML 2.0.

The Users/Groups dialog box opens.

The consumers/Service Providers are the "users" included in the listed user stores.

- d. Click Add/Remove on the appropriate tab.
- e. From the Available Members list, choose the affiliate domains that should have access to Federation Web Services then move them to the Current Members list.

- f. Click OK to return to the Policy List.
- g. Repeat this procedure for all policies relevant for the SAML version you are using.

Federation Web Services is now protected from unauthorized access.

If you try to access Federation Web Services from a link, such as <http://idp-fws.ca.com:81/affwebservices/assertionretriever>, you should be challenged. Only an authorized affiliate site should have access to Federation Web Services.

For this link you must enter a fully-qualified host name and port number for the server where the Federation Web Services application is installed.

To respond to the authentication challenge, enter a valid affiliate name and the affiliate password that has been configured at the Policy Server. These will serve as the credentials.

## Set up a Key Database for Signing POST Responses

To sign SAML POST responses, which is required by the SAML specification, you have to add a private key and certificate to the SiteMinder key database file, `smkeydatabase`.

### More Information:

[Manage the Key Database for Signing and Encryption](#) (see page 479)

## Create Links to Target Resources (optional)

Go to one of the following:

- [Links for SAML 1.x Single Sign-On](#) (see page 190)
- [Links for SAML 2.0 Single Sign-On at the Identity Provider](#) (see page 192)
- [Links to Initiate WS-Federation Single Sign-on](#) (see page 192)

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

At the 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 then sends a request to the producer-side Web Agent before being redirected to a consumer site.

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

For 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 machine 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 machine hosting the Web Agent Option Pack or the SPS federation gateway, depending on which components are installed in your federation network.

For 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 machine 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 machine hosting the Web Agent Option Pack or the SPS federation gateway, depending on which components are installed in your federation network.

**Note:** The SMCONSUMERURL and AUTHREQUIREMENT parameters are not used by SAML POST profile; however, if you include one of these parameters in the intersite transfer URL, you must also include the other.

**More Information:**

[Create Links to Consumer Resources for Single Sign-on](#) (see page 254)

## 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), an unsolicited response at the Identity Provider needs to be initiated. To initiate an unsolicited response, the Federation Web Service application and assertion generator accept an HTTP Get request with a query parameter that indicates the Service Provider ID for which the IdP will generate 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.

You may need to add the [ProtocolBinding query parameter](#) (see page 324) 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 324)

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

To initiate WS-Federation single sign-on, a user clicks on a page with hard-coded HTML link that directs the user's browser 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 that is hosting the Web Agent Option Pack or the SPS federation gateway, depending on which component is installed in your federation network.

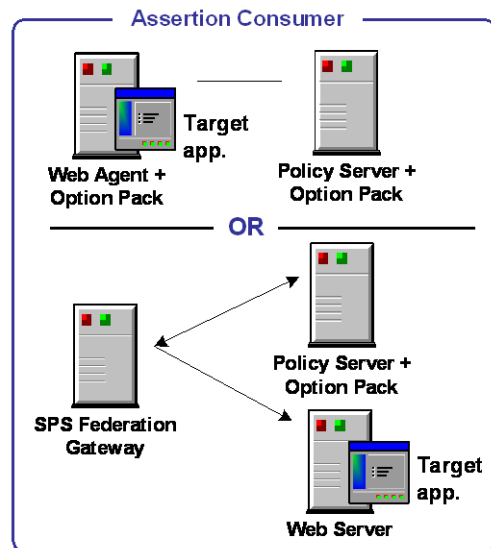
**Note:** You do not need to HTTP-encode the query parameters.

## Set Up Consuming Authority Components

Many of the steps for setting up a Policy Server and Web Agent at the consuming authority are similar to those for the producing authority, with the exception of the following:

- 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.



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

**Note:** This procedure assumes that the target resources already exist at the consuming authority Web site.

## Install the Consuming-side Policy Server

You must install the Policy Server at the consuming authority site. The Policy Server, together with the Policy Server Option Pack provides functions such as the federation authentication schemes and the Assertion Generator.

At the consuming authority, do the following:

1. Install the Policy Server.  
*See SiteMinder Policy Server Installation Guide.*
2. Set up a policy store.  
*See SiteMinder Policy Server Installation Guide.*
3. Set up a user store and add users permitted to access target resources.  
*See SiteMinder Policy Design.*

## Install the Consuming-side Policy Server Option Pack

The Policy Server must already be installed prior to installing the Option Pack.

**Note:** The Policy Server Option Pack requires a JDK. For the supported JDK version, log on to the [Technical Support site](#) and search for the SiteMinder 6.0 Platform Support Matrix.

At the consuming authority, do the following:

1. Install the Policy Server Option Pack.  
*See Policy Server, Web Agent Option Pack Release Notes.*  
**Note:** Be sure to import the affiliate data during the installation. The import is successful if the FederationWebServices domain object is listed in the Domains List in the Policy Server User Interface.
2. Make sure that the NETE\_JRE\_ROOT environment variable is set and that it points to the JVM.

## Configure a SAML or WS-Federation Authentication Scheme

At the consuming authority Policy Server, configure an authentication scheme (artifact, POST profile, SAML 2.0, WS-Federation) for each producing authority.

**Important!** The name of the partner that you specify for the authentication scheme must match the name of the consuming authority that you specify at the producing authority, as follows:

- For SAML 1.x authentication schemes, the Affiliate Name field of the scheme's 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's configuration must match the Resource Partner ID at the Account Partner.

### More Information:

[Configure SiteMinder as a SAML 1.x Consumer](#) (see page 271)

[Configure SiteMinder as a SAML 2.0 Service Provider](#) (see page 353)

[Configure SiteMinder as a Resource Partner](#) (see page 431)

## Protect Target Resources (Consuming-side)

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 consuming authority that require an assertion for user access. Target resources are those identified by the TARGET variable in the intersite transfer URLs (SAML 1.x) or the AuthnRequest URL, or authentication scheme (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 (Consuming-side)

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 consuming authority, set up the following components:

1. Install one of the following:
  - Web Agent  
For instructions, see *SiteMinder Web Agent Installation Guide* and *SiteMinder Web Agent Guide*
  - SPS federation gateway  
For instructions, see *SiteMinder 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 (Consuming-side)

If you are implementing Federation Security Services with a Web Agent and Web Agent Option Pack (not with an SPS federation gateway), you have to install the Web Agent Option Pack. To install this component you need a web or application server.

At the consuming authority:

1. Install one of the following servers to run Federation Web Services, the application 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 Consuming-side 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 consuming authority, do the following:

1. Install the Web Agent Option Pack.

For instructions, see the *Policy Server, Web Agent Option Pack Release Notes*.

2. Be sure to install a JDK, which is required by the Web Agent Option Pack.

To determine the required JDK version, go to the [Technical Support site](#) and search for SiteMinder 6.0 Platform Matrix.

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

## Configure Federation Web Services (Consuming-side)

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 consuming authority

1. Configure a web server running ServletExec or 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. Check that the AgentConfigLocation parameter in the AffWebServices.properties file is set to the full path to the WebAgent.conf file. Ensure 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 producing authority and the consuming authority.
  - 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`

where *fqhn* is the fully-qualified host name and *port\_number* is 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 should see a message that reads:

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 will get 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 \(Producing-side\)](#) (see page 187)

[Federation Web Services Application Setup](#) (see page 211)

## Protect Federation Web Services (Consuming-side)

The procedure for protecting the Federation Web Services application is the same at the consuming authority as it is for the producing authority.

**More Information:**

[Protect Federation Web Services \(Producing-side\)](#) (see page 188)

[Federation Web Services Application Setup](#) (see page 211)

## 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.

If you are implementing artifact single sign-on, smkeydatabase at the producing authority holds the certificate authority's certificate for establishing an SSL connection from the consuming authority to the web server at a producing authority. This 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.

### More Information:

[Manage the Key Database for Signing and Encryption](#) (see page 479)

## Create Links to Initiate Single Sign-on (optional)

For SAML 2.0 and WS-Federation, if a user visits the consuming authority before visiting the producing authority, you have to establish hard-coded links that redirect the user to the producing authority site to fetch the authentication context, which consists of the characteristics that enable the consuming authority to understand how the user was authenticated.

### More Information:

[Initiate SAML 2.0 Single Sign-on at the SP \(optional\)](#) (see page 200)

[Initiate WS-Federation Single Sign-on at the Resource Partner](#) (see page 201)

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

If a user visits the Service Provider first (POST or artifact binding) before visiting the Identity Provider, you have to create an HTML page at the Service Provider that contains hard-coded links to the Service Provider's AuthnRequest Service, which in turn redirects the user to the Identity Provider to fetch the authentication context. The page with the HTML link to the Identity Provider 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 are supported by an HTTP GET request to the AuthnRequest service at the Service Provider's Policy Server.



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 that is hosting the Web Agent Option Pack or the SPS federation gateway.

**IdP\_ID**

Specifies the identity assigned to the Identity Provider

You may need to add the ProtocolBinding query parameter to this link depending on which bindings are enabled. For details on configuring links at the Service Provider and a sample link, 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, you have to create an HTML page, such as a site selection page that contains a list of the Account Partners with which to authenticate. Upon selecting a link, the user is directed to the Single Sign-on Service at the Account Partner. The site selection page has to reside in an unprotected realm.

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

The syntax for the link is:

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

**host:port**

Indicates the server and port number where the Single Sign-on service resides

**RP\_ID**

Specifies the Resource Partner identity

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



# Chapter 5: Setup the SAML 1.x Assertion Generator File

---

This section contains the following topics:

[SAML 1.x Assertion Generator Properties File](#) (see page 203)

## SAML 1.x Assertion Generator Properties File

The Policy Server and Policy Server Option Pack installed at the producer, includes a component called the assertion generator. The assertion generator creates SAML assertions, which are XML documents that contain authentication information about a user.

For the SAML artifact profile, after an assertion is generated, it is stored by the session server until it is requested by the consumer.

The AMAssertionGenerator.properties file (SAML 1.x only) is required for operation of the Assertion Generator. It contains parameters that the Assertion Generator uses to generate SAML 1.x assertions. This file 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 you should change these for your network.

### More Information:

[Configure the SAML 1.x AMAssertionGenerator.properties File](#) (see page 203)

[The JVMOptions.txt File](#) (see page 205)

## 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:** It is essential that this value be properly set so that SAML 1.x assertions are meaningful.

**SecurityDomain**

Identifies the producer's domain, 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 web site](#).

The values you enter in this file should match the values for the equivalent settings at the consumer site, whether the consumer is a SAML Affiliate Agent or a 1.x consumer.

**Note:** If you make any changes to the AmAssertionGenerator.properties file, the changes will not be picked up by the Policy Server until it is restarted.

# Chapter 6: Review the JVMOptions File Used to Create a JVM

---

This section contains the following topics:

[The JVMOptions.txt File](#) (see page 205)

## 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. This file is used by SAML 1.x, SAML 2.0, and WS-Federation.

**Note:** In most situations, you will not need to change this file.

The installed location of this file is:

*policy\_server\_home/config/*

**Important!** If you make changes to the JVMOptions.txt file, you must restart the Policy Server for the changes to take effect. Restart the server from the command line or from the Policy Server Management Console.

To stop and start the Policy Server, use the Policy Server Management Console.

Notes:

- In some environments, when you log off from a system where the SiteMinder Policy Server service is running, the Policy Server service fails because of a JVM issue. To prevent this issue from occurring, 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, you may need to 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 Server](#) (see page 207)

[Configure and Enable the Session Server](#) (see page 208)

[Environments that Require a Shared Session Store](#) (see page 208)

## Federation Data Stored in the Session Server

The Session Server stores data for the following federation features:

- SAML artifact authentication--when artifact authentication is used (SAML 1.x or 2.0), the assertion generator produces a SAML assertion along with an associated artifact that identifies the generated assertion. The artifact is returned to the consumer/Service Provider, while the assertion is stored by the Session Server until the artifact is used to retrieve the assertion.

**Note:** SAML POST profile authentication does not store assertions in the Session Server.

- Single logout--with SAML 2.0 single logout enabled, information about the user's session is stored in the Session Server by the assertion generator and the authentication scheme. When a single logout request is completed, the user's session information is removed from the session store.
- Sign-out --with WS-Federation sign-out enabled, the WS-Federation authentication scheme puts some context information into the Session Server so that a Sign-Out request can be generated. When a signout request is completed, the user's session information is removed from the session store.
- Single use policy--A single use policy is enabled for SAML 2.0 and for WS-Federation by a storage mechanism called expiry data, which is time-based data about the assertion stored by the authentication scheme in the Session Server. Expiry data storage ensures that a SAML 2.0 POST or WS-Federation assertion is only used a single time.

## Configure and Enable the Session Server

You enable the Session Server 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. Choose Session Server from the Database drop-down list.
2. Choose an available storage type from the Storage drop-down list.
3. Set the Enable Session Server option.

You should only enable the Session Server if you are going to use persistent sessions in one or more realms; when enabled, the Session Server impacts Policy Server performance.

**Note:** The Use Policy Store database check box 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

SiteMinder's implementation of the following features requires a session store to store SAML assertions and user session information:

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

The assertion is stored in the session store until it is retrieved by the consumer (Producer/Identity Provider). A persistent session is required for this to work.

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

The single use policy feature prevents SAML 2.0 assertions that arrive via the POST binding from being re-used at a Service Provider to establish a second session. Time-based data about the assertion, known as expiry data, is stored by the authentication scheme in the session store at the Service Provider/Resource Partner. This data ensures that a SAML 2.0 POST or WS-Federation assertion is only used a single time. Although a session store is required at the Service Provider/Resource Partner, a persistent session is not required.

- Single logout (SAML 2.0)

For single logout, the status of the user's session in the session store must be changed to invalidate the session. A persistent session is required at the Identity Provider and Service Provider.



- Signout (WS-Federation)

For WS-Federation signout, the status of the user's session in the session store must be changed to invalidate the session. A persistent session is required at the Account Partner and Resource Partner.

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

- Log-in realm must be configured for persistent sessions for all features *except* for a HTTP-POST single use policy

Persistent sessions are part of the realm configuration.

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

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

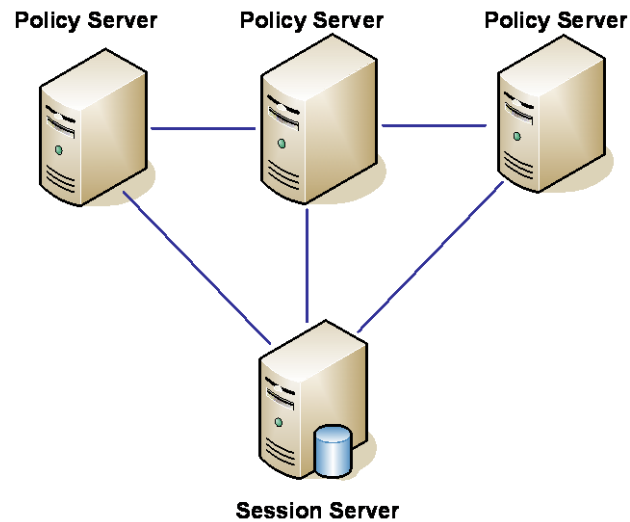
- For SAML 2.0 single logout, and WS-Federation signout, the session store must be shared at the Identity Provider/Account Partner site and the Service Provider/Resource Partner site across all Policy Servers in the cluster.

Sharing the session store ensures 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, the session store must be shared at the consumer site (Service Provider/Resource Partner) across all Policy Servers in the cluster.

**Note:** All Policy Servers that can generate or consume assertions or process a persistent SMSESSION cookie need to be able to contact the common session store. For example, if a user logs into example.com and gets a persistent session cookie for that domain, every single Policy Server that is handling requests for example.com must be able to check that the session is still valid.

The following figure shows a Policy Server cluster communicating with one session server:



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

- Point all Policy Servers to one session server  
In the Policy Server Management Console, configure the Policy Server to use the designated session server.
- Replicate the session store across many session servers  
For instructions on replicating a database, use the documentation for your database.

# Chapter 8: Federation Web Services Application Setup

---

This section contains the following topics:

[Federation Web Services Application Overview](#) (see page 211)

[Configure the Federation Web Services Properties Files](#) (see page 213)

[Deploy Federation Web Services as a Web Application](#) (see page 217)

[Protect the Federation Web Services Application](#) (see page 234)

[Flush Federation Web Services Cache for Trace Logs](#) (see page 238)

## Federation Web Services Application Overview

Federation Web Services, a component of Federation Security Services, is a collection of servlets packaged as a Web application in accordance with the Java Servlet API 2.3 specification. It is installed with the Web Agent Option Pack or deployed inside the Tomcat web server, which is embedded in the SPS federation gateway.

The web application is rooted at a specific URL within the web server. For example, the application could be located at <http://www.myserver.com/affwebservices/> and the assertion retrieval service (part of the application) could be registered to listen on <http://www.myserver.com/affwebservices/assertionretrieval>. All requests that start with this URL prefix are routed to the assertion retrieval servlet.

A web application exists as a structured hierarchy of directories. The root of the hierarchy serves as a document root for serving files that are part of this context.

A special directory, WEB-INF, exists within the application directory hierarchy, located at one of the following locations:

- *web\_agent\_home*/affwebservices/WEB-INF (Web Agent/Web Agent Option Pack)
- *sps\_home*/secure-proxy/Tomcat/webapps/affwebservices/WEB-INF (SPS federation gateway)

**web\_agent\_home**

Indicates the installed location of the Web Agent

**sps\_home**

Indicates the installed location of the Secure Proxy Server

This directory contains everything related to the application that is not in the document root of the application. That is, the WEB-INF directory is not part of the public document tree of the application, and no file contained in the WEB-INF directory may be served directly to the client.

The contents of the WEB-INF directory are:

- /WEB-INF/classes/: Properties files, such as LoggerConfig.properties.
- /WEB-INF/lib/: Directory for Java archive files containing servlets, beans, and other utility classes that are useful to the application.
- Web.xml: lists the servlets and URL mappings.

The Federation Web Services application provides these services:

- Assertion Retrieval Service (SAML 1.x)
- SAML credential collector to consume assertions for single sign-on (SAML 1.x)
- Intersite Transfer Service (SAML 1.x)
- Artifact Resolution Service (SAML 2.0)
- Assertion Consumer Service for single sign-on (SAML 2.0)
- Security Token Consumer Service for single sign-on (WS-Federation)
- AuthnRequest service (SAML 2.0)
- Single Sign-on service for cross-domain single sign-on (SAML 2.0 and WS-Federation)
- Single Logout Service (SAML 2.0)
- Signout Service (WS-Federation)

- Session Synchronization -- ValidateSession & Logout calls--a value-added service, supported by a standards-based SOAP RPC mechanism (used by the SAML Affiliate Agent only).
- Notification Alert--a value-added service, supported by a standards-based SOAP RPC mechanism (used by the SAML Affiliate Agent only)

**Note:** The Session Synchronization and Notification Alert services can only be used when the Web Agent/Web Agent Option Pack is at the producing site and the SAML Affiliate Agent is at the consuming site. These services are not supported with the SPS federation gateway.

## Federation Web Services Deployment Descriptors

There are two deployment descriptor files used by Federation Web Services:

- DeployedServices.ds--this binary file lists all the services that are exposed. This file is used by the SOAP toolkit. You should not have to change this file.
- Web.xml--lists the servlets and URL mappings. You should not need to change this file, but you can modify the URL mappings.

### More Information:

[Federation Web Services URLs Used in SiteMinder Configuration](#) (see page 623)

## Configure the Federation Web Services Properties Files

The Federation Web Services application relies on the AffWebServices.properties file for its initialization parameters. You need to familiarize yourself with this file and configure some of its parameters.

## Configure the AffWebServices.properties File

The AffWebServices.properties file contains all the initialization parameters for Federation Web Services.

The installed location of the AffWebServices.properties file is in one of two locations:

- For a Web Agent:  
*web\_agent\_home/affwebservices/WEB-INF/classes*
- For the SPS federation gateway:  
*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 Secure Proxy Server

### **To configure the AffWebServices.properties file**

1. Set the AgentConfigLocation parameter to the location of the WebAgent.conf file at the consumer and producer sites.
  - Windows example for the Web Agent:  
*C:\Program Files\ca\webagent\bin\IIS\WebAgent.conf*  
**Note:** Federation Web Services is a Java component, so the Windows paths must contain double back-slashes. This does not apply for UNIX platforms.
  - UNIX example for the Web Agent:  
*SunJavaSystem\_server/servers/https-hostname/config/WebAgent.conf*
  - Windows example for SPS federation gateway  
*sps\_home\proxy-engine\conf\defaultagent\WebAgent.conf*
  - UNIX example for SPS federation gateway  
*sps\_home/proxy-engine/conf/defaultagent/WebAgent.conf*

Repeat this procedure for each application server installed with the Web Agent Option Pack.

- For the rest of the settings, you can accept the default values or modify as needed.

| AffWebServices.properties Settings | Value                                                                                                                               |
|------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------|
| NotificationLibraryType            | Specifies the library type the Web Agent uses for notification alerts.<br><b>Note:</b> Not supported by the SPS federation gateway. |
| NotificationLibraryDetails         | Indicates the Java classname or the C library and function name.<br><b>Note:</b> Not supported by the SPS federation gateway.       |
| SMserverPort                       | Determines which Policy Server service at the producer processes the notification tunnel calls.                                     |
| AgentConfigLocation                | Indicates the location of the WebAgent.conf file. If you are using a 4.x IIS or Sun ONE Web Agent, this field can be left blank.    |

## Set up the LoggerConfig.properties File

The LoggerConfig.properties file lets you enable logging so the Federation Web Services application can record assertion retrieval, session management, notification alert information, and trace messages related to Federation Web Services.

The log file may show activity at the producing authority and the consuming authority, depending on how your site is configured.

The installed location of the LoggerConfig.properties file is:

- For the Web Agent, the location is  
*web\_agent\_home/affwebservices/WEB-INF/classes*
- For the SPS federation gateway:  
*sps\_home/secure-proxy/Tomcat/webapps/affwebservices/WEB-INF/classes*

**web\_agent\_home**

Indicates the installed location of the Web Agent

**sps\_home**

Indicates the installed location of the Secure Proxy Server

Modify the settings as needed. If a value is not specified, the default value for the default locale is used.

The following table shows the settings in the LoggerConfig.properties file.

| LoggerConfig.properties Settings | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
|----------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| EnabledDNSLookup                 | Instructs the FWS application whether or not to do a DNS or reverse DNS lookup when processing an incoming SAML request at the consuming site. Select Y or N.<br><br>When an incoming SAML request is received at a consumer site, FWS logs the details of the request, including the requesting host name. The host name is collected by the DNS look up call.<br><br>The default behavior is to do the DNS lookup. If you select N for this heading, the DNS call is not made and the IP address is logged instead. |
| LoggingOn (required)             | Enables log output. Select Y or N.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| LocalFileName (required)         | Names the file to use for log output                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| LogLocalTime                     | Enables use of local time for log messages. Select Y or N.                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| LogRollover                      | Defines the type of rollover functionality. Select Y or N then define the LogSize or LogCount parameter.                                                                                                                                                                                                                                                                                                                                                                                                              |
| LogSize                          | Specifies the maximum file size, in megabytes, when rolling over log files by size.                                                                                                                                                                                                                                                                                                                                                                                                                                   |



| LoggerConfig.properties Settings | Description                                                                                                                                  |
|----------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------|
| LogCount                         | Specifies how many log output files to leave when rollover is enabled.                                                                       |
| TracingOn                        | Enables trace log output. Select Y or N.                                                                                                     |
| TraceFileName                    | Names the file to use for trace log output.                                                                                                  |
| TraceConfig                      | Specifies the trace configuration file. For more information, see <a href="#">Federation Security Services Trace Logging</a> (see page 471). |
| TraceRollover                    | Defines the type of rollover functionality for tracing. Select Y or N and then specify a TraceSize or TraceCount value.                      |
| TraceSize                        | Specifies the maximum file size, in megabytes, when rolling over trace log files by size.                                                    |
| TraceCount                       | Specifies how many trace log output files to leave when rollover is enabled.                                                                 |
| TraceFormat                      | Specifies the trace output file format (default, fixed width fields, delimited format, XML)                                                  |
| TraceDelim                       | Defines the character to use as a delimiter when using fixed width fields as the trace format.                                               |

## Deploy Federation Web Services as a Web Application

If you are using the Web Agent Option Pack, you have to deploy the Federation Web Services application into operation.

Follow the appropriate instructions for your environment:

- [Configure ServletExec to work with Federation Web Services](#) (see page 218)
- [Configure WebLogic Application Server to work with Federation Web Services](#) (see page 222)

- [Configure WebSphere Application Server to work with Federation Web Services](#) (see page 226)
- [Configure JBOSS or Tomcat Application Server to work with Federation Web Services](#) (see page 231)

For the SPS federation gateway, Federation Web Services is already deployed on the embedded Tomcat server.

## Configure ServletExec to Work with Federation Web Services

For the Federation Web Services application to work with ServletExec, you need to specify Federation Web Services as a Web application for ServletExec and add several JVM options for both the producing authority and the consuming authority.

Note: You should have received a ServletExec license key file called licensekey50.txt from CA 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> web site.

Remember, ServletExec and the Web Agent Option Pack must be installed on the same web server where you installed the Web Agent.

**Important!** Be sure to apply the most current hot fixes for ServletExec. Without the hot fixes, Federation Web Services will not work with ServletExec. To obtain the hot fixes, go to the [New Atlanta Communication web site](#).

### To set up ServletExec to work with FWS

1. Open the ServletExec Administration Console.
2. Under Web Applications, select manage.  
The Manage Web Applications dialog opens.
3. Click Add a Web Applications.
4. Enter the following information:
  - a. Application Name: *affwebservices*
  - b. URL Context Path: */affwebservices/*
  - c. Location: *affwebservices\_home*  
Example:  
`C:\program files\ca\webagent\affwebservices`
5. Click Submit.

6. (Optional) For better performance, add the following Virtual Machine options:
  - a. Under Virtual Machine, select options.  
The Java Virtual Machine (VM) Options dialog opens.
  - b. Add the following Java VM Options. If there is not sufficient space to add all four options together, add then submit each option separately.  
  
`-Djavax.xml.parsers.SAXParserFactory=org.apache.xerces.  
jaxp.SAXParserFactoryImpl`  
  
`-Dorg.apache.xerces.xni.parser.XMLParserConfiguration=org.apache.xerces.  
parsers.XML11Configuration`  
  
`-Dorg.xml.sax.driver=org.apache.xerces.parsers.SAXParser`  
  
`-Djavax.xml.parsers.DocumentBuilderFactory=org.apache.xerces.  
jaxp.DocumentBuilderFactoryImpl`
7. Click Submit after you have added all four entries.
8. Exit the ServletExec Console.

## Set the ServletExec Library Path Variable

You need to set the library path to deploy FWS on a ServletExec server. Set this path in the in the StartServletExec.bat file (Windows) or StartServletExec shell script (UNIX) for the appropriate ServletExec instance.

The variable name for the library path differs depending on the operating system:

- Solaris: LD\_LIBRARY\_PATH
- HP-UX: SHLIB\_PATH
- LINUX: LD\_LIBRARY\_PATH
- AIX: LIBPATH

### To set the library path on Machine 2 and Machine 5

1. Open a command window.
2. Open the appropriate file in an editor.
3. Set the variable.

Example: LD\_LIBRARY\_PATH=/webagent\_option\_pack\_home/bin

## Modify the AffWebServices.properties File for ServletExec

The [AffWebServices.properties](#) (see page 214) file contains all the initialization parameters for FWS. When deploying FWS, be sure you have modified the AgentConfigLocation setting to point to the WebAgent.conf file.

## Enable ServletExec to Write to the IIS File System

The IIS Web server does not allow a plug-in to write to its file system unless it is configured with a user account that has proper rights to do so. Therefore, for ServletExec to write to the federation log files, the anonymous user account that you associate with ServletExec must have permissions to write to file system.

### To enable the user account used by ServletExec to write to the IIS file system

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 group box.  
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. Refer to Windows documentation to grant this right to a user account. For example, you might use the IUSR Internet Guest account for anonymous access.
  - b. Deselect Basic authentication.
  - c. Deselect 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 associated with ServletExec can now write to the IIS file system.

Additionally, you must give the anonymous user the right to act as part of the operating system.

To give the anonymous user account the right to act as part of the operating system

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.

Although it is optional, we strongly recommend that you, look at the Agent Configuration Object for the Web Agent protecting the IIS Web server and ensure that the SetRemoteUser parameter is set to yes to make sure that the anonymous user can write to the file system.

## Ensure the IIS Default Web Site Exists

The Web Agent requires the IIS Web server to have a Default Web Site for proper installation. The Default Web Site is automatically installed with the IIS Web server. If this Web site does not exist, or you wish to install the SiteMinder virtual directories to a different Web site on IIS, you need to edit the Metabase.

A technical note on the [site](#) describes the [Technical Support site](#) changes that are needed. To find the note:

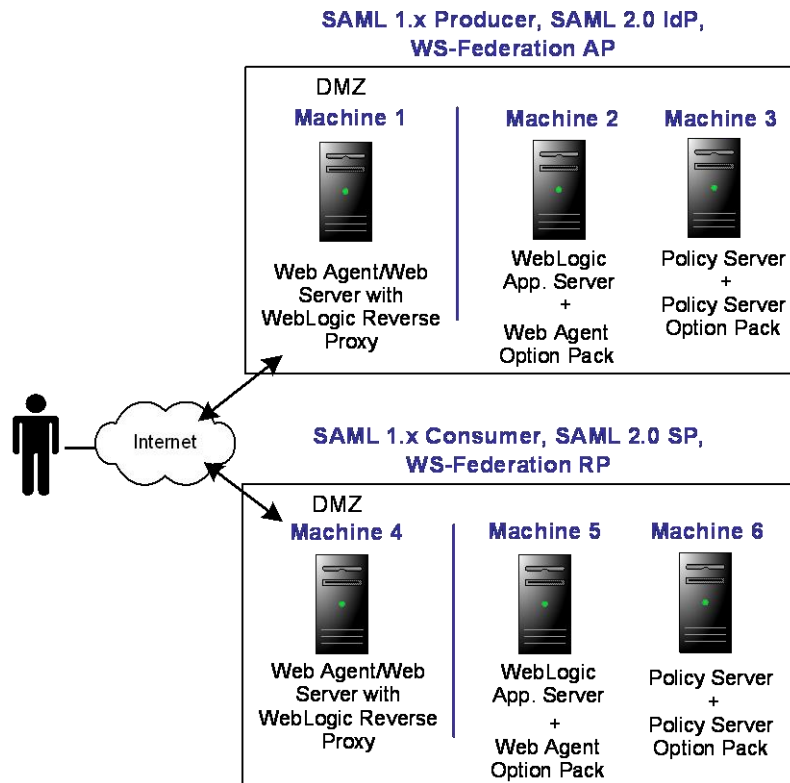
1. Go to the main Support page.
2. Select Literature,Tech Notes.
3. Select the document titled METABASE -3 Error.

The documents are listed in alphabetical order.

## Configure WebLogic to Work with Federation Web Services

To enable Federation Web Services (FWS) for a SiteMinder/WebLogic configuration, deploy the FWS application.

The following illustration shows a SiteMinder and WebLogic sample configuration. It provides an example of how to use FWS in a sample federated environment.



In this environment, deploy the FWS application on Machine 2 and Machine 5.

**Important:** Complete the deployment procedure for the Web Agent at the producing authority site and the consuming authority site.

After installing the software components on the machines shown in the previous illustration, deploy the FWS application on Machine 2 for the producing authority site and on Machine 5 for the consuming authority site by following this procedure:

1. Set the LD\_LIBRARY\_PATH Variable
2. Create a SmHost.conf File
3. Create a WebAgent.conf File
4. Modify the AffWebServices.properties File

5. Configure the WebLogic Reverse Proxy Plug-in
6. Deploy the FWS Application on WebLogic

**Important!** For the FWS application to work with WebLogic Server 8.1.4 or higher, check the `weblogic.xml` file in the `WEB-INF` directory and make sure the `prefer-web-inf-classes` parameter is set to `true`. The `weblogic.xml` file is located in the following directory:

`webagent\affwebservices\WEB-INF`

The following code excerpt shows how the `prefer-web-inf-classes` parameter should be set:

```
<weblogic-web-app>
  <container-descriptor>
    <prefer-web-inf-classes>true</prefer-web-inf-classes>
  </container-descriptor>
</weblogic-web-app>
```

If `prefer-web-inf-classes` is set to `false`, change the value to `true`.

## Set the WebLogic Library Path Variable

You need to set the library path to deploy FWS on a WebLogic server. Set this path in the `startWebLogic.cmd` file (Windows) or the `startWebLogic.sh` file (UNIX) systems.

The variable name for the library path differs depending on the operating system:

- Solaris: `LD_LIBRARY_PATH`
- HP-UX: `SHLIB_PATH`
- LINUX: `LD_LIBRARY_PATH`
- AIX: `LIBPATH`

### To set the library path on Machine 2 and Machine 5

1. Open a command window.
2. Open the appropriate file in an editor.
3. Set the variable.

Example: `LD_LIBRARY_PATH=/webagent_option_pack_home/bin`

## Create an SmHost.conf File

The FWS application requires an SmHost.conf file; however, the Web Agent Option Pack does not install this file so you need to create it.

### To create an SmHost.conf

1. Go to the directory `/webagent_option_pack_home/bin`
2. Run the `smregghost.exe`.

For instructions on running `smregghost.exe`, see the *SiteMinder Web Agent Installation Guide*.

3. Put the SmHost.conf file in the following directory on Machines 2 and 5:  
`/webagent_option_pack_home/config`

## Create a WebAgent.conf File

The FWS application requires the WebAgent.conf file; however, the Web Agent Option Pack does not install this file so you need to create it.

To create a WebAgent.conf file:

1. Copy the WebAgent.conf file from Machine 1 to the following directory on Machine 2 and Machine 5:

`/webagent_option_pack_home/config`

where `webagent_option_pack_home` is the installed location of the Web Agent Option Pack on Machine 2 or Machine 5.

2. Modify the WebAgent.conf file by:
  - a. Setting the `EnableWebAgent` parameter to YES.
  - b. Modifying other configuration parameters to suit FWS.

The following is a sample of a WebAgent.conf file for the FWS application:

```
# WebAgent.conf - configuration file for the Federation Web Services Application
#agentname="agent_name, IP_address"
HostConfigFile="/webagent_option_pack/config/SmHost.conf"
AgentConfigObject="agent_config_object_name"
EnableWebAgent="YES"
```

**Note:** The Agent Configuration Object referenced in this WebAgent.conf file must be a new object that you create. Do not specify the object in use by the Web Agent installed in your environment.



## Modify the AffWebServices.properties File for WebLogic

The [AffWebServices.properties](#) (see page 214) file contains all the initialization parameters for FWS. When deploying FWS, be sure you have modified the AgentConfigLocation setting to point to the WebAgent.conf file.

## Configure the WebLogic Reverse Proxy Plug-in

To set up the WebLogic Reverse Proxy plug-in:

1. On Machine 1, configure the WebLogic reverse proxy plug-in on the Apache Web Server.

For more information, see WebLogic's documentation.

2. Add the following aliases to the Web server's configuration file.

This example uses the Apache httpd.conf file.

```
<IfModule mod_weblogic.c>
WebLogicHost <WebLogic_Machine_IP_Address>
WebLogicPort <WebLogic_Machine_Port_Number>
</IfModule>

<Location /affwebservices>
SetHandler weblogic-handler
Debug ALL
</Location>
```

## Deploy the FWS Application on WebLogic

To deploy the FWS application on Machine 2 and Machine 5:

1. Use the WebLogic Server Console and deploy FWS. The FWS application is installed in:

*/webagent\_option\_pack\_home/affwebservices/*

For information about deploying a Web application, see WebLogic's documentation.

2. Test that the Federation Web Services application is working. Open a Web browser and enter:

*http://fqhn:port\_number/affwebservices/assertionretriever*

where *fqhn* is the fully-qualified host name and *port\_number* is 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 should see a message that reads:

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.  
The FWS application is now deployed for the WebLogic server.

If Federation Web Services is not operating correctly, you will get a message that the Assertion Retrieval Service has failed. In case of failure, check the Federation Web Services log.

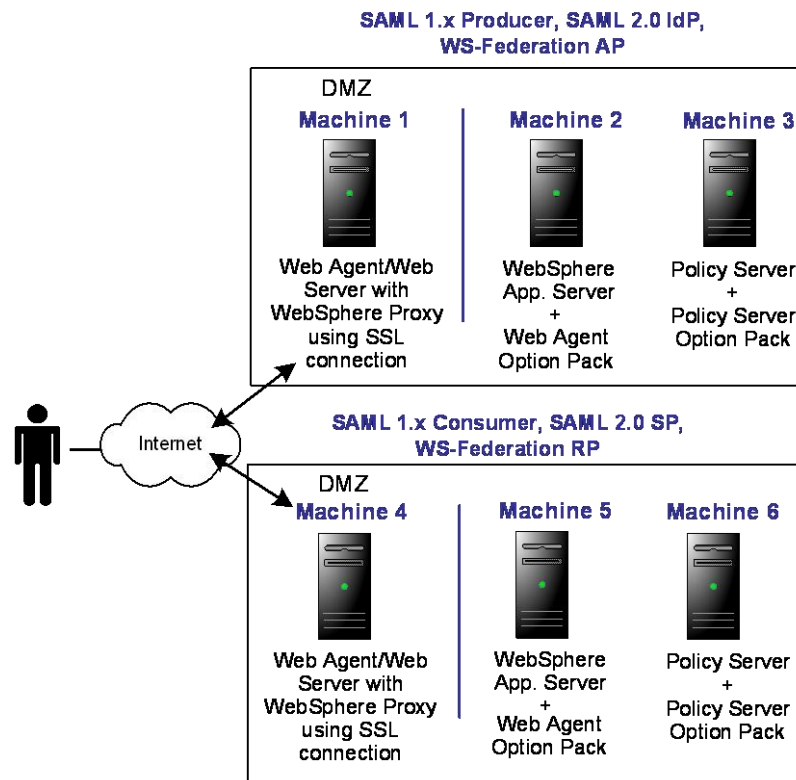
**Note:** For instructions on enabling trace logging for the FWS application, see [Federation Security Services Trace Logging](#) (see page 471).

## Configure WebSphere to Work with Federation Web Services

To enable FWS in a federated environment for a SiteMinder/WebSphere Application Server (WAS) configuration, deploy the FWS application.

On Machines 2 and 5, deploy FWS. These machines must also have WAS and the associated WAS Fix Pack, if applicable. On Machines 1 and 4, install the Web Agent and the WAS Proxy Plug-in. Enable SSL between the Proxy and the WAS.

The following illustration shows a SiteMinder and WebSphere sample configuration.



#### Prerequisites:

- WAS must be installed on systems that have the WebSphere Application Server installed for Federation Web Services to work.
- Complete the deployment procedure for the Web Agent at the producer/IdP site and the consumer/SP site.

After installing the software components on the machines listed in the previous illustration, deploy this application on Machine 2 and Machine 5 by following these steps:

Step 1: Set the WebSphere LD\_LIBRARY\_PATH Variable

Step 2: Create a SmHost.conf File

Step 3: Create a WebAgent.conf File

Step 4: Modify the AffWebServices.properties File

Step 5: Copy Option Pack Library Files to WebSphere

Step 6: Deploy a Federation Web Services WAR File in WebSphere

## Set the WebSphere Library Path Variable

You need to set the library path to deploy FWS on WebSphere. Set this path in the startServer.bat file (Windows) or the startServer.sh file (UNIX) systems.

The variable name for the library path differs depending on the operating system:

- Solaris: LD\_LIBRARY\_PATH
- HP-UX: SHLIB\_PATH
- LINUX: LD\_LIBRARY\_PATH
- AIX: LIBPATH

### To set the library path on Machine 2 and Machine 5

1. Open a command window.
2. Open the appropriate file in an editor.
3. Set the library path variable.

Example: LD\_LIBRARY\_PATH=/webagent\_option\_pack\_home/bin

## Create an SmHost.conf File

The FWS application requires the SmHost.conf file; however, the Web Agent Option Pack does not install this file so you need to create it.

### To create an SmHost.conf file

1. Create an SmHost.conf file by running smregghost.exe, which is located in  
`/webagent_option_pack_home/bin`  
For instructions on running smregghost.exe, see the *SiteMinder Web Agent Installation Guide*.
2. Put the SmHost.conf file in the following directory on Machine 2 and Machine 5:  
`/webagent_option_pack_home/config`

## Create a WebAgent.conf File

The FWS application requires the WebAgent.conf file; however, the Web Agent Option Pack does not install this file so you need to create it.

To create a WebAgent.conf file:

1. Copy the WebAgent.conf file from Machine 1 to the following directory on Machine 2 and Machine 5:

*/webagent\_option\_pack\_home/config*

where *webagent\_option\_pack\_home* is the installed location of the Web Agent Option Pack on Machine 2 and Machine 5.

2. Modify the WebAgent.conf file by:
  - a. Setting the EnableWebAgent parameter to YES.
  - b. Modifying any other configuration parameters to suit the environment for the FWS application.

The following is a sample of a WebAgent.conf file for the FWS application:

```
# WebAgent.conf - configuration file for the Federation Web Services Application
#agentname=<agent_name>, <IP_address>
HostConfigFile="/<webagent_option_pack>/config/SmHost.conf"
AgentConfigObject=<agent_config_object_name>
EnableWebAgent="YES"
```

**Note:** The Agent Configuration Object referenced in this WebAgent.conf file must be a new object that you create. Do not specify the object in use by the Web Agent installed in your environment.

## Modify the AffWebServices.properties File for WebSphere

The [AffWebServices.properties](#) (see page 214) file contains all the initialization parameters for FWS. When deploying FWS, be sure you have modified the AgentConfigLocation setting to point to the WebAgent.conf file.

## Copy Web Agent Option Pack Libraries to WebSphere

To copy the Web Agent Option Pack library files on Machine 2 and Machine 5:

1. Copy the following files from the directory *\webagent\_option\_pack\bin*
  - smcommonutil.dll
  - smerrlog.dll
  - smfedclientcomponent.dll

- smjavaagentapi.dll
  - libetpki\_openssl\_ssl.a (AIX only)
  - libetpki\_openssl\_crypto.a (AIX only)
2. Place the copied libraries in the following directory:  
`\WebSphere_home\AppServer\bin`

## Deploy a Federation Web Services WAR File in WebSphere

To deploy a FWS WAR file on Machine 2 and Machine 5:

1. Create a WAR file of the Federation Web Services application. The application is installed in:

`\webagent_option_pack\affwebservices\`

For instructions on creating a WAR file, see WebSphere's documentation.

2. Deploy the WAR file using WebSphere Administrator's Console.

For instructions, see WebSphere's documentation.

**Important!** Any subsequent changes made to any of the properties files in the `affwebservices` directory requires you to re-create a WAR file and re-deploy this file in the application server.

3. From the WebSphere Administrator's Console, go to Applications, Enterprise Applications.
4. Select the name of the web services WAR file, such as `affwebservices_war`.
5. On the Configuration tab:
  - a. Set the Classloader Mode to `PARENT_FIRST`.
  - b. Set WAR Classloader Policy to `Application`.
  - c. Save the settings.
6. Test that the Federation Web Services application is working by opening a Web browser and entering:

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

where *fqhn* is the fully-qualified host name and *port\_number* is 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 should see a message that reads:

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

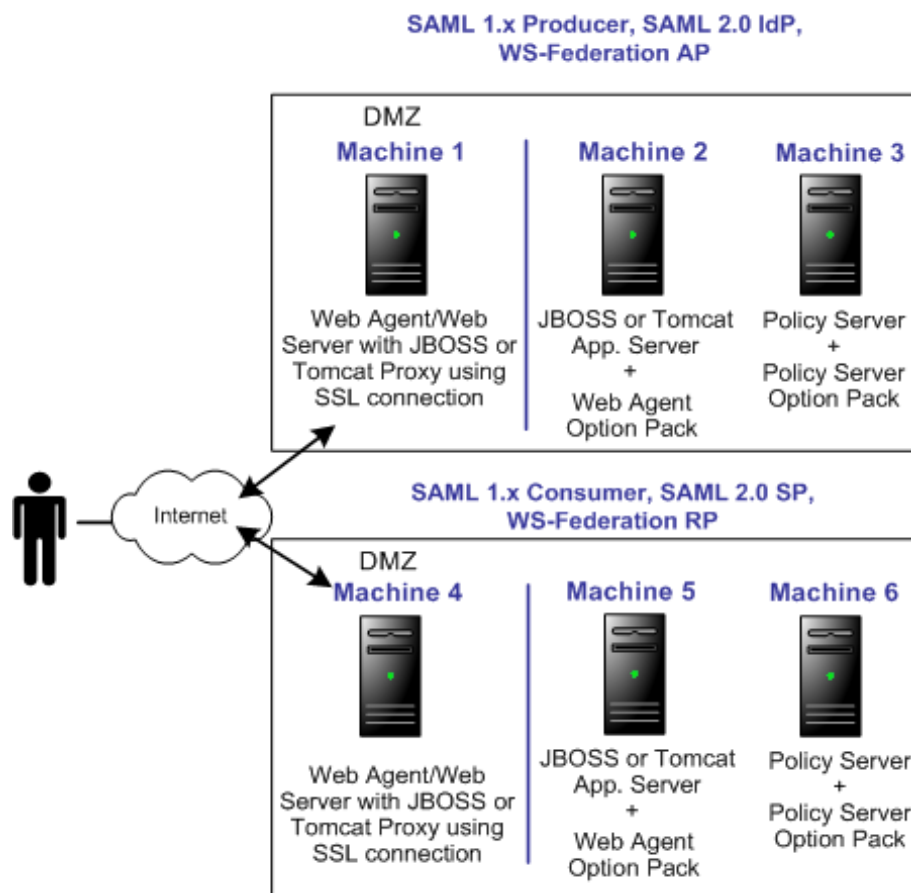
This message indicates that Federation Web Services is listening for data activity. If Federation Web Services is not operating correctly, you will get a message that the Assertion Retrieval Service has failed. In case of failure, check the Federation Web Services log.

**Note:** For instructions on enabling trace logging for the FWS application, see [Federation Security Services Trace Logging](#) (see page 471).

## Configure JBOSS or Tomcat to Work with Federation Web Services

To use a JBoss or Tomcat Application Server in a SiteMinder federated environment, deploy the FWS application on the application server.

The following picture shows the deployment with JBOSS or Tomcat. On Machines 1 and 4, the Web Agent is installed with the JBOSS or Tomcat Connector for proxy support. SSL is enabled between the proxy and the application server. On Machines 2 and 5, FWS is deployed with the application server by way of the Web Agent Option Pack. On Machines 3 and 6, FWS is deployed with the application server by way of the Policy Server Option Pack.



The process for deploying FWS is as follows:

1. Create an SmHost.conf file.
2. Create a WebAgent.conf file.
3. Modify the AffWebServices properties file.
4. Deploy the FWS WAR file in the application server.
5. Modify the Tomcat catalina.properties file (Tomcat 6.0.18 and higher)

The following sections detail each step in the process.

### Create an SmHost.conf File

The FWS application requires the SmHost.conf file; however, the Web Agent Option Pack does not install this file so you must create it.

1. Create an SmHost.conf file by running smregghost.exe, which is located in  
`/webagent_option_pack/bin`  
For instructions on running smregghost.exe, see the *SiteMinder Web Agent Installation Guide*.
2. Put the SmHost.conf file in the following directory on Machine 2 and Machine 5:  
`/webagent_option_pack/config`

### Create a WebAgent.conf File

The FWS application requires the WebAgent.conf file; however, the Web Agent Option Pack does not install this file so you must create it.

1. Copy the WebAgent.conf file from Machine 1 to the following directory on Machine 2 and Machine 5:  
`/webagent_option_pack/config`  
where *webagent\_option\_pack* is the installed location of the Web Agent Option Pack on Machine 2 and Machine 5.



2. Modify the WebAgent.conf file as follows:
  - a. Set the EnableWebAgent parameter to YES.
  - b. Modify any other configuration parameters to suit the environment for the FWS application.

The following is a sample of a WebAgent.conf file for the FWS application:

```
# WebAgent.conf - configuration file for the Federation Web Services Application
#agentname="agent_name, IP_address"
HostConfigFile="/web_agent_home/config/SmHost.conf"
AgentConfigObject="agent_config_object_name"
EnableWebAgent="YES"
```

**Note:** The Agent Configuration Object referenced in this WebAgent.conf file must be a new object that you create. Do not specify the object in use by the Web Agent installed in your environment.

## Modify the AffWebServices.properties File for JBOSS or Tomcat

The [AffWebServices.properties](#) (see page 214) file contains all the initialization parameters for FWS. When deploying FWS, be sure you have modified the AgentConfigLocation setting to point to the WebAgent.conf file.

## Deploy an FWS WAR File in JBoss or Tomcat

### To deploy the FWS application for a JBOSS or Tomcat server

1. Open a command window and navigate to the affwebservices directory, located in:  
*/webagent\_option\_pack/affwebservices/*.

2. Create a WAR file by entering the command:

```
jar cvf affwebservices.war *
```

For information about deploying a Web application, see the documentation for your application server.

3. Copy the affwebservices.war file to the appropriate server location:

#### **JBOSS**

*JBOSS\_home/server/default/deploy/*

*JBOSS\_home* is the installed location of the JBOSS application server.

#### **Tomcat**

*Tomcat\_home/server/default/deploy/*

*Tomcat\_home* is the installed location of the Tomcat application server.

4. Restart the application server.

5. After the server has restarted, access the JBOSS or Tomcat Administrative Console. All the services supported by affwebservices are displayed on the main Console page.
6. Test that the FWS application is working by opening a web browser and entering the following:

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

**fqhn**

Represents the fully-qualified host name and

**port\_number**

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

For example:

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

If FWS is operating correctly, the following message is displayed:

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

This message indicates that FWS is listening for data activity. The FWS application is now deployed for the application server.

If FWS is not operating correctly, you receive a message that the Assertion Retrieval Service has failed. If there is a failure, review the FWS log.

**Note:** For instructions on enabling trace logging for the FWS application, see [Federation Security Services Trace Logging](#) (see page 471).

## Protect the Federation Web Services Application

When you install the Policy Server and Policy Server Option Pack, policies for each service that comprises the Federation Web Services application are automatically created. You must enforce the protection of the Federation Web Services application using SiteMinder policies.

Optionally, you may have to protect the Assertion Retrieval Service (SAML 1.x) or the Artifact Resolution Service (SAML 2.0) for artifact authentication. These services are components of Federation Web Services.

### More Information:

[Enforce Policies that Protect Federation Web Services](#) (see page 235)

[Protect the Assertion Retrieval or Artifact Resolution Service \(optional\)](#) (see page 236)

## Enforce Policies that Protect Federation Web Services

The Federation Web Services (FWS) application is protected by SiteMinder policies.

When you install the Policy Server and Policy Server Option Pack, these policies and the related policy objects are automatically created by the `ampolicy.smdif` file. There is one policy for each service that makes up the Federation Web Services application.

The following table lists the objects and policies that protect FWS.

| Object Type           | Object Name                                                                                                                                          |
|-----------------------|------------------------------------------------------------------------------------------------------------------------------------------------------|
| Domain                | FederationWebServicesDomain                                                                                                                          |
| Realm                 | FederationWebServicesRealm<br>public                                                                                                                 |
| Agent Group           | FederationWebServicesAgentGroup                                                                                                                      |
| Rule                  | FederationWSAssertionRetrievalServiceRule<br>FederationWSNotificationServiceRule<br>FederationWSSessionServiceRule<br>SAML2FWSArtifactResolutionRule |
| Policy                | FederationWSAssertionRetrievalServicePolicy<br>FederationWSNotificationServicePolicy<br>SAML2FWSArtifactResolutionServicePolicy                      |
| User Context Variable | AllowNotification                                                                                                                                    |
| User Context Variable | AllowSessionSync                                                                                                                                     |
| User Directory        | FederationWSCustomUserStore<br>SAML2FederationCustomUserStore                                                                                        |

You must enforce protection of the Federation Web Services policies by adding the Web Agent protecting these services to an Agent group. All other aspects of configuring the policies, such as the Basic authentication scheme, realms and rules are set up automatically. Additionally, you must specify the affiliates/Service Providers who can access the Federation Web Services application. Additionally, you need to permit the affiliates access to the Federation Web Services application.

### To enforce policies for the Federation Web Services application

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

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

2. Specify the affiliates who are permitted to access the Federation Web Services application. This requires adding affiliates, Services Providers, or Resource Partners as users to the appropriate policies in the FederationWebServicesDomain.

**Note:** You have to establish affiliate domains and add affiliates to the domains prior to giving the affiliates permission.

### More Information:

[Configure SiteMinder as a SAML 1.x Producer](#) (see page 243)

[Configure SiteMinder as a SAML 2.0 Identity Provider](#) (see page 301)

[Configure SiteMinder as an Account Partner](#) (see page 405)

## Protect the Assertion Retrieval or Artifact Resolution Service (optional)

If you configure an artifact authentication scheme at a consumer/Service Provider, the Assertion Retrieval Service (SAML 1.x) and the Artifact Resolution Service (SAML 2.0) are the mechanisms that retrieve the assertion from the Policy Server at the producer/Identity Provider.

We strongly recommend that you protect the Assertion Retrieval Service (SAML 1.x) and the Artifact Resolution Service (SAML 2.0) against unauthorized access.

To protect these services, you specify an authentication scheme for the realm that contains the service at the producer/Identity Provider. The authentication scheme dictates the type of credentials that the SAML credential collector (SAML 1.x) or the Assertion Consumer Service (SAML 2.0) must provide to access the relevant service.

You can select one of the following authentication schemes:

- Basic over SSL
- X.509 client certificate

### Use Basic over SSL Scheme to Protect the Assertion Retrieval Service

To use a Basic over SSL scheme to protect the Assertion Retrieval Service (SAML 1.x) and the Artifact Resolution Service (SAML 2.0), no additional configuration is required at the producer/Identity Provider. A set of default policies are already configured when you install the Policy Server and Policy Server Option Pack.

At the consumer/Service Provider, there is also no configuration required, provided you can use one of the default root Certificate Authorities (CAs) already in the smkeydatabase, which is used to establish an SSL connections between the consumer/Service Provider and the producer/Identity Provider. If you want to use your own root CA instead of a default CA, you have to import the CA certificate into the smkeydatabase.

**More Information:**

[Enforce Policies that Protect Federation Web Services](#) (see page 235)

## Use a Client Cert. to Protect the Assertion Retrieval or Artifact Resolution Service

To use a client certificate authentication scheme, you:

1. Create a policy at the producer/Identity Provider to protect the relevant service. This policy will use the client certificate authentication scheme.
2. Enable client certificate authentication at the consumer/Service Provider.

## How to Use Client Cert. Authentication with an IIS 5.0 Web Server

Client certificate authentication is not supported for IIS 5.0 Web servers at the producer/Identity Provider. However, it can be used on an IIS 5.0 Web server at the consumer/Service Provider to communicate with a non-SiteMinder producer/Identity Provider.

To work around this issue, use the IIS 5.0 Web server's client certificate functionality at the producer/Identity Provider and do not configure SiteMinder's client certificate functionality. If you apply this workaround, be aware that the CN portion of the certificate's DN value must contain the affiliate name value.

**More Information:**

[Protect the Artifact Resolution Service with Client Certificate Authentication \(optional\)](#)  
(see page 345)

[Protect the Assertion Retrieval Service with Client Certificate Authentication \(optional\)](#)  
(see page 267)

[Configure the Client Certificate Option at the Consumer](#) (see page 277)

## Flush Federation Web Services Cache for Trace Logs

If you modify any part of the federation configuration at the producing authority or the consuming authority, for example, you enable or disable POST binding, you need to flush the Federation Web Services cache for the changes to appear in the trace logs.

### To flush the cache

1. Log on to the Policy Server User Interface.
2. Select Tools, Manage Cache  
The Cache Management dialog opens.
3. Click Flush All in the All Caches group box.
4. Click OK.

All caches will now be cleared.

# Chapter 9: Creating Affiliate Domains

---

This section contains the following topics:

[Affiliate Domain Overview](#) (see page 239)

[Configure an Affiliate Domain](#) (see page 240)

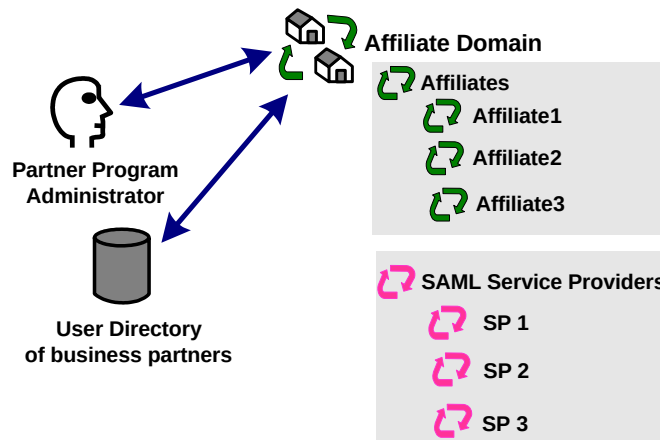
## Affiliate Domain Overview

An affiliate domain is a logical grouping of federated entities 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 authenticate a user, SiteMinder 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 affiliate domain's search order.

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.

The following illustration shows a partner program administrator for the affiliate domain that can manage properties for each consumer. The marketing administrator does not have these privileges.



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 privilege Manage Affiliates. A system administrator that can grant control over a policy domain to other administrators has the privilege Manage System and Domain Objects.

### More Information:

[Assign User Directories](#) (see page 241)

## 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 240)
2. [Assign User Directories](#) (see page 241)
3. [Assign an Administrator](#) (see page 241)
4. [Add Entities to an Affiliate Domain](#) (see page 242)

## Add a Domain Object

### To configure an affiliate domain

1. Log into the Policy Server User Interface.
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.

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 Policy Server User Interface 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 Policy Server User Interface 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 producing authority when you [protect the Federation Web Services application](#) (see page 235).

For instructions on adding consuming authorities 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 243)

[Configure SiteMinder as a SAML 2.0 Identity Provider](#) (see page 301)

[Configure SiteMinder as an Account Partner](#) (see page 405)

# Chapter 10: Configure SiteMinder as a SAML 1.x Producer

---

This section contains the following topics:

- [Prerequisites for Producing SAML 1.x Assertions](#) (see page 243)
- [Configuration Checklist for 1.x Producer](#) (see page 244)
- [Add a Consumer to an Affiliate Domain](#) (see page 245)
- [Select Users for Which Assertions Will Be Generated](#) (see page 247)
- [Configure SAML 1.x Assertions to Authenticate Users](#) (see page 250)
- [Create Links to Consumer Resources for Single Sign-on](#) (see page 254)
- [Allow Access to the Federation Web Services Application](#) (see page 256)
- [Set Up Sessions for a SAML Affiliate Agent Consumer \(optional\)](#) (see page 257)
- [Configure Attributes to Include in SAML 1.x Assertions \(Optional\)](#) (see page 259)
- [Configure IP Address Restrictions for 1.x Consumers \(optional\)](#) (see page 262)
- [Configure Time Restrictions for 1.x Consumers \(optional\)](#) (see page 263)
- [Customize SAML 1.x Assertion Content \(optional\)](#) (see page 263)
- [Protect the Authentication URL to Create a SiteMinder Session \(SAML 1.x\)](#) (see page 265)
- [Protect the Assertion Retrieval Service with Client Certificate Authentication \(optional\)](#) (see page 267)

## Prerequisites for Producing SAML 1.x Assertions

To produce SAML 1.x assertions for consumers, the following conditions must be met:

- The Policy Server and Policy Server Option Pack must be installed. (The Policy Server and Policy Server Option Pack installs the Assertion Generator and SAML authentication schemes).
- The session server, a component of the Policy Server must be 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.

- Install one of the following options:
  - The Web Agent and the Web Agent Option Pack on a Web server. You need a Web Agent to authenticate a user and establish a SiteMinder session. You need the Option Pack to install 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 must be set up within the federated network. The SAML assertions generated at the Policy Server must be forwarded to an application that can receive and interpret the assertions. The SAML Affiliate Agent and the SAML Credential Collector (installed with the Web Agent Option Pack) can both act as SAML consumers (1.0 and 1.x respectively).

## Configuration Checklist for 1.x Producer

At the producer-side Policy Server, you add consumers to an affiliate domain and define aspects of the consumers' configuration so the producer can issue assertions to each one.

See the following:

- [Required Configuration Tasks](#) (see page 245)
- [Optional Configuration Tasks](#) (see page 245)

Tips:

- Certain parameter values at the producer and consumer 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 503).
- To ensure 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 623).

**Note:** If you installed a SAML Affiliate Agent as part of the SiteMinder Federation Services solution, the SAML Affiliate Agent configuration takes place only at the consumer site, not at the producer-side Policy Server. The SAML Affiliate Agent is a separately-licensed product.

## Required Configuration Tasks at a 1.x Producer

The tasks required to identify a consumer to the producer are as follows:

- Establish an affiliate domain.
- Add consumers to the affiliate domain. These consumers must have permission to access Federation Web Services at the producer.
- Select users for which the Producer will generate assertions.
- Configure assertions.

### More Information:

[Add a Consumer to an Affiliate Domain](#) (see page 245)

[Select Users for Which Assertions Will Be Generated](#) (see page 247)

[Configure SAML 1.x Assertions to Authenticate Users](#) (see page 250)

[Create Links to Consumer Resources for Single Sign-on](#) (see page 254)

## 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:

[Set Up Sessions for a SAML Affiliate Agent Consumer \(optional\)](#) (see page 257)

[Configure Attributes to Include in SAML 1.x Assertions \(Optional\)](#) (see page 259)

[Configure IP Address Restrictions for 1.x Consumers \(optional\)](#) (see page 262)

[Configure Time Restrictions for 1.x Consumers \(optional\)](#) (see page 263)

[Customize SAML 1.x Assertion Content \(optional\)](#) (see page 263)

## 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 Policy Server User Interface.
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 prior to redirecting the user to establish a SiteMinder session. Encrypting the SMPORTALURL protects it from being modified by a malicious user.

**Note:** If you select this checkbox, 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:**

[Protect the Authentication URL to Create a SiteMinder Session \(SAML 1.x\)](#) (see page 265)

[SAML 1.x Affiliate Dialog Reference](#) (see page 509)

## Select Users for Which Assertions Will Be Generated

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 Policy Server User Interface 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 Policy Server User Interface 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 SAML 1.x Assertions to Authenticate Users

Administrators at a producer site determine how the Policy Server packages SAML assertions for delivery to a consumer. The assertion document contains user and session information that serves as a user's credentials for authentication purposes.

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

- Information about the consumer
- Session information
- User attributes

For complete information about SAML assertions, refer to the SAML specification at the [OASIS web site](#).

For details on how an assertion is consumed when a SAML credential collector is installed at the consumer, see the single sign-on diagrams for SAML 1.x in [Federation Security Services Process Flow](#) (see page 85).

**Note:** For assertion retrieval when a SAML Affiliate Agent is the consumer, see the *SiteMinder SAML Affiliate Agent Guide*.

## A Security Issue Regarding SAML 1.x Assertions

The SAML assertion generator creates an assertion based on a session for a user that has been authenticated at any authentication scheme protection level. This presents a security issue--you can control which users an assertion is generated for, but not based on the protection level at which they authenticated.

You may have resources that should be accessed only by users who have authenticated at a particular protection level. If your site's resources are secured at different protection levels, ensure that when users authenticate to establish a session, they do so with the desired protection level to ensure the federated environment's security.

## Configuring a SAML 1.x Assertion

The Assertions tab lets you define how assertions are sent to the consumer. The assertion is used to authenticate the user at the consumer site.

### To configure a SAML 1.x assertion

1. Log into the Policy Server User Interface.
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 SAML 1.x artifact binding, a value specified for the Assertion Consumer URL takes precedence over the value of the SMCONSUMERURL query parameter, which is a required query parameter of the intersite transfer URL that a user selects to initiate single sign-on. If you are concerned that the SMCONSUMERURL could be manipulated to send the user to an unauthorized site where the artifact is retrieved, we recommend specifying 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:**

[Affiliate Dialog--Assertions Tab](#) (see page 514)

## Assertion Validity for Single Sign-on

For single sign-on, the values of the Skew Time and the Validity Duration determine how SiteMinder FSS calculates the total time that an assertion is valid. FSS 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 the NotBefore and NotOnOrAfter values represent the beginning and end of the validity interval.

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

**NotBefore=IssueInstant - Skew Time**

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

### Calculating Assertion Validity if FSS is at Both Sides of the Partnership

If FSS is at both sides of a partnership, the total time the assertion is valid 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 FSS, 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 takes the NotBefore and NotOnOrAfter values from the assertion and applies its skew time to those values to calculate new NotBefore and NotOnOrAfter values.

Skew Time = 180 seconds (3 minutes)

NotBefore = 4:56PM

NotOnOrAfter=5:05PM

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

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

## Create 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, which makes a request to the producer-side Web Agent before the user is redirected to the consumer site.

For 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 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 web site where the user is authenticated

**affiliate\_name**

Indicates the name of an affiliate configured in an affiliate domain.

**consumer\_site**

Indicates the site 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. These parameters are supported by an HTTP GET request to the producer Web Agent.

**Note:** Query parameters for the SAML artifact profile must use HTTP-encoding.

| Query Parameter           | Meaning                                                                     |
|---------------------------|-----------------------------------------------------------------------------|
| SMASSERTIONREF (required) | For internal use. The value will always be QUERY. Do not change this value. |

| Query Parameter                                           | Meaning                                                                                                                                                                                                                                    |
|-----------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| NAME<br>(required)                                        | Name of an affiliate configured in an affiliate domain.                                                                                                                                                                                    |
| TARGET<br>(required)                                      | The target URL at the consumer site.                                                                                                                                                                                                       |
| SMCONSUMERURL (required only for artifact profile)        | The URL at the consumer site processes the assertion and authenticates the user.<br><br>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<br>(required only for artifact profile) | For internal use. The value will always be 2. Do not change this value.                                                                                                                                                                    |

**Note:** The SMCONSUMERURL and AUTHREQUIREMENT parameters are not used by SAML POST profile; 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 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 Policy Server User Interface.

**Note:** To set up persistent sessions, configure the session server. Set up a session server using the Policy Server Management Console.

## Allow Access to the Federation Web Services Application

After you add affiliates to an affiliate domain, the affiliates need permission to access the Federation Web Services application. When you install the Policy Server and Policy Server Option Pack, the FederationWebServicesDomain is installed by default. This domain includes the following policies:

- FederationWSAssertionRetrieval
- FederationWSNotificationService
- FederationWSSessionServicePolicy
- SAML2FWSArtifactResolutionServicePolicy

### To specify permission to the Federation Web Services application

1. From the Domains tab, expand FederationWebServicesDomain and select Policies.
2. Select one of the policies, and click Edit, Properties of Policy.

For SAML 1.x, you need to permit access to:

- FederationWSAssertionRetrieval
- FederationWSNotificationService
- FederationWSSessionServicePolicy

For SAML 2.0, you need to permit access to SAML2FWSArtifactResolutionServicePolicy

The SiteMinder Policy dialog opens.

3. From the Users tab, select one of the following:
  - FederationWSCustomUserStore tab for SAML 1.x
  - SAML2FederationCustomUserStore tab for SAML 2.0.

The Users/Groups dialog opens.

The consumers, Service Providers, and Resource Partners are the "users" included in the listed user stores.

4. Click Add/Remove on the appropriate tab.



5. From the Available Members list, choose the affiliate domains that should have access to Federation Web Services then move them to the Current Members list.
6. Click OK to return to the Policy List.
7. Repeat this procedure for all policies relevant for the SAML version you are using.

## Set 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:

- Session management should be 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 SAML POST profile. Sessions can only be used with SAML artifact profile.

Session management between a producer and a SAML Affiliate Agent may 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 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 sessioning model, the producer and the SAML Affiliate Agent can maintain a shared session. Sessions at the producer and the SAML Affiliate Agent are terminated if the producer session expires or the user logs out at either site.

**Note:** For more information about session management, see the *SiteMinder 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 258)

## Set the Sync Interval for Shared Sessions

The *sync interval* defines the frequency, while the user is active at the consumer, 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 sent by the producer.)

The sync interval ensures that the information at the producer's session server and the information in the SAML Affiliate Agent cookies is synchronized. For example, if 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 based only on the value of the sync interval; the user has to be active at the consumer—that is, the user is requesting consumer resources.

Two considerations affect the value of Sync Interval:

- If the value of Sync Interval is too small, the SAML Affiliate Agent keeps calling the session server, which slows down SiteMinder's performance when processing requests.
- 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 at the producer before going to the SAML Affiliate Agent. The Idle Timeout Enabled field is part of a realm's session configuration parameters.

**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)

Attributes can be included in assertions for use by servlets, web, or custom applications to display customized content for a user or enable other custom features. 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 offer fine-grained access control by limiting what a user can do at the consumer. For example, you can send an attribute named Authorized Amount and set it to a maximum dollar amount that the user can spend at the consumer.

Attributes take the form of name/value pairs and include information, such as the user's mail address or business title, or an approved spending limit for transactions at the consumer. When the consumer receives the assertion, it takes the attributes and makes them 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, Federation Security Services 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, Federation Security Services 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 Affiliates dialog.

### More Information:

[Affiliate Dialog--Attributes Tab](#) (see page 516)

## Attribute Types

Attributes identify the information that the Policy Server passes to the consumer. There are several types of attributes that you can use in an attribute. The types of attributes determine where the Policy Server finds the proper attribute values.

You can specify the following types of attributes:

- **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.

**Note:** For the Policy Server to return response attributes that contain values from user directory attributes, the user directories must be configured in the SiteMinder 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 Policy Server to return response attributes using values from DN attributes, the user directories must be configured in the SiteMinder User Directory dialog box.

When you configure a response attribute, the correct Value Type for the response attribute is displayed in the upper right corner of the SiteMinder Response Attribute Editor dialog box.

## 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.

#### More Information:

[Affiliate Dialog--Attributes Tab](#) (see page 516)

### 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 Policy Server User Interface 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. If IP addresses have been specified for a consumer, only users who access the consumer site from the appropriate IP addresses will be accepted by the consumer.

The procedure for specifying IP address restrictions for a consumer is the same as configuring IP restrictions for a policy in a policy domain.

**More Information:**

[Affiliate Dialog--IP Addresses Tab](#) (see page 520)

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

The Policy Server User Interface allows 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.

**More Information:**

[Affiliate Dialog--Time Tab](#) (see page 520)

## Customize SAML 1.x Assertion Content (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 box.

**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 reference 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*.

**More Information:**

[Affiliate Dialog--Advanced Tab](#) (see page 521)

## Integrate the Assertion Generator Plug-in with SiteMinder (SAML 1.x)

After writing an assertion generator plug-in, you have to integrate the plug-in with SiteMinder.

The instructions for compiling the SAML 1.x assertion plug-in Java file are in the `AssertionSample.java` file, in `sdk/samples/assertiongeneratorplugin`.

To integrate the assertion generator plug-in with SiteMinder:

1. Compile the assertion plug-in Java file.

This file requires the following .jar files 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 enables the plug-in to be loaded.

**Note:** Do not modify the classpath for xercesImpl.jar, xalan.jar, or SMJavaApi.jar.

3. Restart the Policy Server.

Restarting ensures that the latest version of the assertion plug-in is picked up after being recompiled.

Instead of specifying the assertion plug-in class and its parameters via the Policy Server User Interface you can use the Policy Management API (C or Perl). For instructions, see the *SiteMinder Programming Guide for C* or the *Javadoc Reference*, which is part of the *SiteMinder Programming Guide for Java*.

## Protect the Authentication URL to Create a SiteMinder Session (SAML 1.x)

When you add a consumer to an affiliate domain, one of the parameters you are required to set is the AuthenticationURL parameter.

The file that the AuthenticationURL points to is the redirect.jsp file. This file is installed at the producer 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 the Web Agent presents an authentication challenge to users who request a protected consumer resource but do not have a SiteMinder session.

A SiteMinder session is required for the following features:

- For users requesting a protected Service Provider resource

If you configure single sign-on using an HTTP artifact profile, a persistent session is needed to store SAML assertions in the session server.

- For single sign-on using an HTTP POST profile

A user must have a session, but it does not have to be a persistent session because assertions are delivered directly to the consumer site through the user's browser. The assertions do not have to be stored in the session server.

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 so that the Agent can process the request and generate 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 Policy Server User Interface.
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 247).

## Protect the Assertion Retrieval Service with Client Certificate Authentication (optional)

By default, there is a pre-configured policy that uses the Basic over SSL authentication scheme to protect the Assertion Retrieval Service. When you configure the policy for the client certificate authentication scheme, you create this policy for a different realm than the realm that uses the Basic over SSL scheme.

Generally, the administrator at the Identity Provider should create two policies to protect the Assertion Retrieval Service by Basic over SSL and to protect it with client certificate authentication.

To protect the Assertion Retrieval Service using a client certificate authentication scheme, you:

- Create a policy at the producer that uses an X.509 client certificate authentication scheme.
- Enable client certificate authentication at the consumer.

### More Information:

[Create the Assertion Retrieval Service Policy](#) (see page 268)

[Access the Assertion Retrieval Service with a Client Certificate \(optional\)](#) (see page 276)

## Use of Client Cert. Auth. with an IIS 5.0 Web Server

Client certificate authentication is not supported for IIS 5.0 Web servers at the producer/Identity Provider. However, it can be used on an IIS 5.0 Web server at the consumer/Service Provider to communicate with a non-SiteMinder producer/Identity Provider.

To work around this issue, use the IIS 5.0 Web server's client certificate functionality at the producer/Identity Provider and do not configure SiteMinder's client certificate functionality. If you apply this workaround, be aware that the CN portion of the certificate's DN value must contain the affiliate name value.

## Create the Assertion Retrieval Service Policy

### To create a policy that protects the Assertion Retrieval Service

1. For each affiliate, add an entry to a user directory. You can create a new user store or use an existing directory.

Create a separate user record for each affiliate site that retrieving assertions from the producer site.

An attribute of the user record should have the same value that is specified in the Name field of the Affiliate Properties dialog box.

For example, if you identified the affiliate as Company A in the Name field, the user directory entry should be:

`uid=CompanyA, ou=Development, o=CA`

The Policy Server will map the subject DN value of the affiliate's client certificate to this directory entry.

2. Add the configured user directory to the FederationWebServicesDomain.
3. Create a certificate mapping entry.

The value for the Attribute Name field in the Certificate Mapping Properties dialog box should be mapped to the user directory entry for the affiliate. The attribute represents the subject DN entry in the affiliate's certificate. For example, you may select CN as the Attribute Name, and this represents the affiliate named `cn=CompanyA, ou=Development, o=CA`

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
- Authentication Scheme: client cert authentication scheme created in the previous step.

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

- 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 make sure that the affiliate site for which the SAML assertion was generated is the site actually 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. For example, the entry could be uid=CompanyA.

Based on the following entries, the Web Agent will return 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*
9. Complete the configuration steps at the Service Provider to use client certificate authentication, if they are not completed already.

Instructions can be found in [Access the Assertion Retrieval Service with a Client Certificate \(optional\)](#) (see page 276).

# Chapter 11: Configure SiteMinder as a SAML 1.x Consumer

---

This section contains the following topics:

[SAML 1.x Authentication Schemes](#) (see page 271)  
[Configuration Tasks for SAML 1.x Authentication](#) (see page 276)  
[Access the Assertion Retrieval Service with a Client Certificate \(optional\)](#) (see page 276)  
[SAML 1.x Authentication Scheme Prerequisites](#) (see page 279)  
[Configure SAML 1.x Artifact Authentication](#) (see page 280)  
[Configure SAML 1.x POST Profile Authentication](#) (see page 282)  
[Configure SAML 1.x User Disambiguation](#) (see page 284)  
[Customize Assertion Processing with the Message Consumer Plug-in](#) (see page 286)  
[Supply SAML Attributes as HTTP Headers](#) (see page 288)  
[Specify Redirect URLs for Failed SAML 1.x Authentication](#) (see page 293)  
[How To Protect a Resource with a SAML 1.x Authentication Scheme](#) (see page 295)

## SAML 1.x Authentication Schemes

If you purchased the Policy Server and Policy Server Option Pack, any SiteMinder site can consume SAML 1.x assertions and use these assertions to authenticate and authorize users. If you have sites in your federated network that have user stores, you may want to use SAML authentication.

There are two SAML 1.x authentication methods available for configuration with SiteMinder:

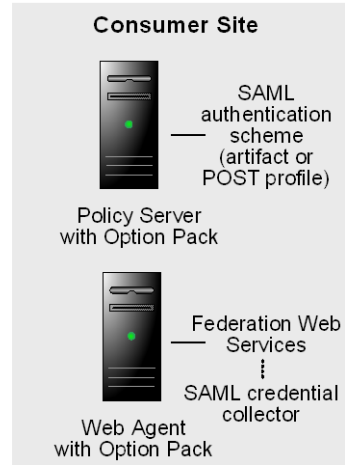
- SAML Artifact profile
- SAML POST profile

The SAML-based authentication schemes let a consumer site in a federated network 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 consumer site can authorize the user for specific resources.

A consumer is a site that uses a SAML 1.x assertion to authenticate a user. A producer is a site that generates SAML 1.x assertions.

**Note:** A site may be both a SAML producer and a SAML consumer.

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 SiteMinder Federation Web Services application functions. For information about installing and configuring the SPS federation gateway, see the *CA SiteMinder Secure Proxy Server Administration Guide*.

The SAML 1.x authentication scheme is configured at the consumer-side Policy Server and is invoked by the SAML credential collector. The SAML credential collector is a component of the Federation Web Services application and is installed on the consumer-side Web Agent or 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 user's credentials to login 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.

**More Information:**

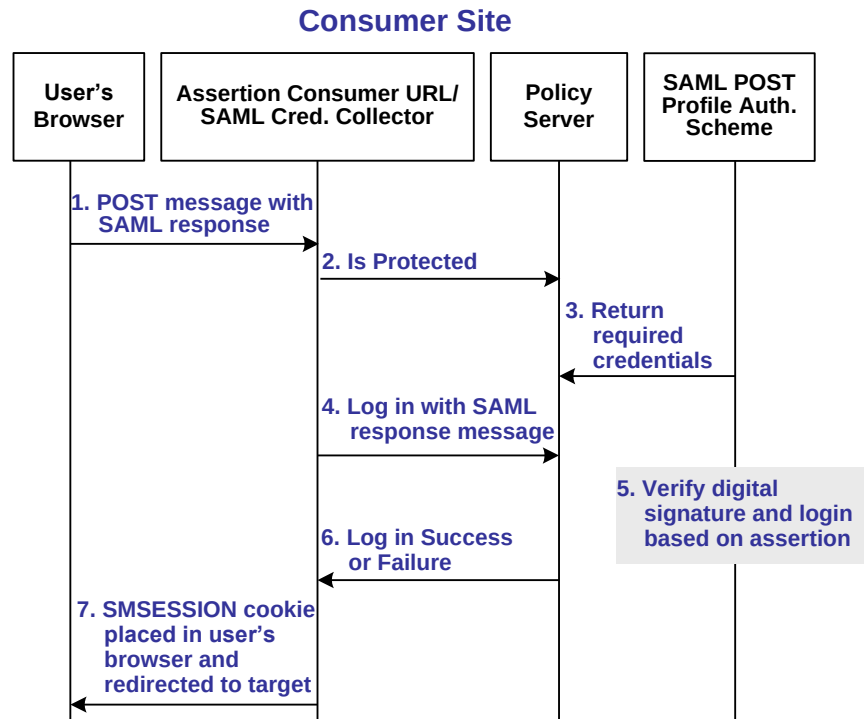
[SAML 1.x Artifact Authentication Scheme Overview](#) (see page 274)

[SAML 1.x POST Profile Authentication Scheme Overview](#) (see page 273)



## SAML 1.x POST Profile Authentication Scheme Overview

The following illustration shows how the SAML POST profile authentication scheme processes requests.



**Note:** The SPS federation gateway or the Web Agent Option Pack can provide the Assertion Consumer Service and SAML Credential Collector functionality necessary for the authentication process.

Unless otherwise stated, the following process takes place at the consumer site:

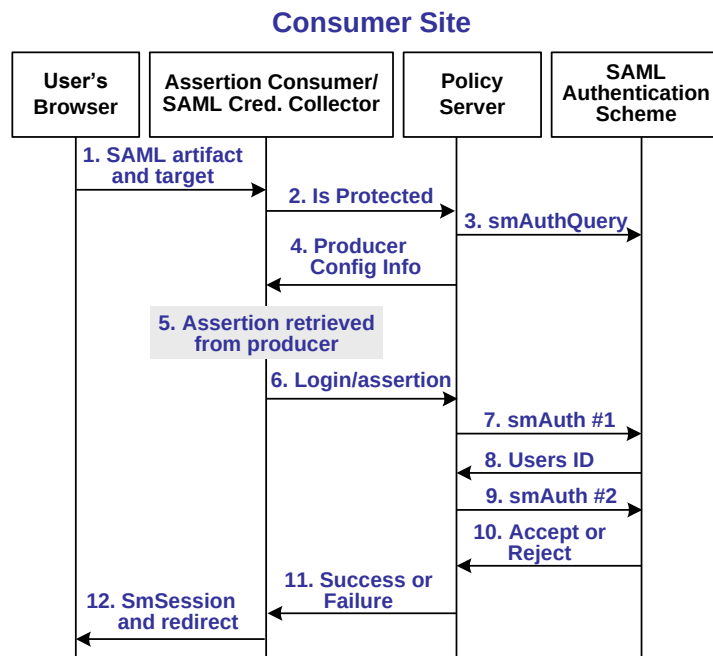
1. A user's browser POSTs an HTML form to the Assertion Consumer URL (which is the URL for the SAML credential collector). This form contains a SAML response message and target URL originally generated at the producer.
2. The SAML credential collector makes a call to the Policy Server to determine if the target resource is protected.
3. The Policy Server replies that the target URL is protected by the SAML POST profile authentication scheme. This indicates to Federation Web Services application that 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. Assuming login succeeds, the SAML credential collector sets an SMSESSION cookie, which it puts in the user's browser, and then redirects the user to the target resource. If the login fails, the credential collector redirects the user to the configured No Access URL.

## SAML 1.x Artifact Authentication Scheme Overview

The following illustration shows how the SAML 1.x artifact authentication scheme processes requests.

This illustration shows the SAML 1.x artifact authentication functional model.



**Note:** The SPS federation gateway or the Web Agent and Web Agent Option Pack can provide the Agent and SAML Credential Collector functionality necessary for the authentication process.

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 is originally generated from the SiteMinder Web Agent at the producer site.
2. The SAML credential collector calls the Policy Server to check if the requested resource is protected. This resource is protected by the SAML artifact authentication scheme.
3. The Policy Server passes the necessary data to the SAML artifact authentication scheme, which extracts the producer configuration information, such as the affiliate name and password.
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 stored at the producer Policy Server.
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 two-step authentication process is the standard authentication process as dictated by the SiteMinder Authentication API. 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. If the login to the Policy Server succeeds, the SAML credential collector creates a session cookie and places it in the user's browser then redirects the user to the target resource. If the login fails, the credential collector servlet redirects the user to a No Access URL.

## Configuration Tasks for SAML 1.x Authentication

The tasks to create a SAML 1.x authentication scheme are listed in the following table.

| Check Here | Task                                                                                                                                                                                 |
|------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|            | Create an authentication scheme for each producer that will generate assertions.                                                                                                     |
|            | Associate that scheme with a realm and add that realm as part of a policy. You can do this on a per producer basis or create a single custom authentication scheme and single realm. |
|            | Optional: For SAML artifact authentication, you can configure client certificate authentication to access the assertion retrieval service.                                           |

Notes:

- Certain parameter values at the producer and consumer 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 503).
- To ensure you are using the correct URLs for the Federation Web Services servlets, review the list of URLs, which can be found in [Federation Web Services URLs Used in SiteMinder Configuration](#) (see page 623).

**More Information:**

[Configure SAML 1.x Artifact Authentication](#) (see page 280)

[Configure SAML 1.x POST Profile Authentication](#) (see page 282)

[How To Protect a Resource with a SAML 1.x Authentication Scheme](#) (see page 295)

[Access the Assertion Retrieval Service with a Client Certificate \(optional\)](#) (see page 276)

## Access the Assertion Retrieval Service with a Client Certificate (optional)

This procedure is for single sign-on only with the artifact profile.

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 Artifact authentication scheme is invoked by the SAML credential collector, which 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. This tells the SAML credential collector 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, there are some configuration tasks at the consumer and the producer that you need to complete, as follows:

- At the consumer, select the client certificate option to indicate that a certificate will be presented as credentials.
- At the producer, create a policy to protect the Assertion Retrieval Service.

**More Information:**

[Configure the Client Certificate Option at the Consumer](#) (see page 277)

[Protect the Assertion Retrieval Service at the Producer](#) (see page 278)

## Configure the Client Certificate Option at the Consumer

Do the following to enable client certificate authentication:

1. [Select the Client Cert Option for Authentication](#) (see page 277)
2. [Add a Client Certificate to the Smkeydatabase](#) (see page 277)

### 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

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 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.

**Notes on Creating a Private Key**

- When you create a private key, the `dname` value, which specifies the consumer name, can be any attribute from the consumer's subject DN because the Policy Server at the producer site can use its certificate mapping functionality to map the consumer to a local user directory entry. This means that in the Certificate Mapping Properties dialog box of the Policy Server User Interface, you can use any of the values listed for the Attribute Name field in the Mapping information. In this example, CN is used; however, you can use other attributes, such as OU, O, UID.

- The value for alias associated with the private key should be same as the value of the Affiliate Name field specified in the Scheme Setup dialog for the SAML Artifact Authentication scheme. The attribute of the consumer's subject DN, represented in the example by the CN value, should also reflect the Affiliate Name value.

For example, if you entered CompanyA as the Affiliate Name, then alias would be Company A, and the attribute could be CN=CompanyA, OU=Development, O=CA, L=Waltham, ST=MA, C=US

- To refer to the existing key store entry, subsequent keytool commands must use the same alias.
- The value for password should be same as the value of the Password field specified in the Scheme Setup dialog for the SAML Artifact Authentication Scheme.

## Protect the Assertion Retrieval Service at the Producer

At the producer-side Policy Server, configure a policy to protect the Assertion Retrieval Service. The realm for this policy must use an X.509 client certificate authentication scheme.

**More Information:**

[Protect the Artifact Resolution Service with Client Certificate Authentication \(optional\)](#)  
(see page 345)

## SAML 1.x Authentication Scheme Prerequisites

There are several prerequisites you must fulfill before configuration a SAML authentication scheme.

### Install the Policy Server and Its Option Pack

The Policy Server together with the Policy Server Option Pack provides the SAML authentication scheme at the consumer. It also provides the SAML assertion generator used by a producer.

Install the Policy Server and the Policy Server Option Pack at the producer and consumer sites.

For installation instructions, refer to the following guides:

- *Policy Server Installation Guide*
- *Policy Server, Web Agent Option Pack Guide*

### Install Federation Web Services at the Producer and Consumer

Federation Web Services (FWS) is a web application. FWS provides the SAML credential collector servlet, which consumes assertions and other services for federated network configurations.

To use the FWS application features, install the Web Agent and Web Agent Option Pack or the SPS federation gateway, which has FWS embedded at the producer and consumer sites.

For installation and configuration instructions, refer to the following:

- *Policy Server and Web Agent Option Pack Guide*
- *Secure Proxy Server Administration Guide*

**Important!** If you install a Web Agent, you must define a value for the Web Agent configuration parameter `DefaultAgentName` for all consumer Web Agents. This value specifies a Web Agent identity. Additionally, the specified Agent identity must be included in the Resource Filter of the realm that protects the target resource. You configure the `DefaultAgentName` parameter in the Agent Configuration Object or the local Agent configuration file. Omitting the `DefaultAgentName` parameter or using the value specified in the `AgentName` parameter in the realm resource filter causes SAML 1.x authentication to fail, regardless of the single sign-on profile.

## Set Up a Key Database to Sign and Verify SAML POST Responses

To use the SAML POST profile for passing assertions, the assertion generator at the producer site needs to sign the SAML response that contains the assertion. The assertion consumer at the consumer site needs to verify that signature.

To accomplish these tasks, you must set up a key database for each Policy Server that is responsible for signing, verification or both. 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.

### More Information:

[Manage the Key Database for Signing and Encryption](#) (see page 479)

## 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 279).
2. Log into the Policy Server User Interface.
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 by configuring the fields on the tab.

**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 503).

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:**

[Authentication Scheme Dialog--SAML Artifact Template](#) (see page 524)

[How To Protect a Resource with a SAML 1.x Authentication Scheme](#) (see page 295)

[Access the Assertion Retrieval Service with a Client Certificate \(optional\)](#) (see page 276)

[Modify the Key Database Using smkeytool](#) (see page 486)

[Authentication Scheme Dialog--SAML Artifact Template--Scheme Setup Tab](#) (see page 525)

## 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

### More Information:

[Authentication Scheme Dialog--SAML Artifact Template--Advanced Tab](#) (see page 532)

## 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 279).
2. Log into the Policy Server User Interface.
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 282)

[Configure a Custom SAML 1.x POST Authentication Scheme](#) (see page 283)

## 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 503).

**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:**

[Authentication Scheme Dialog--SAML POST Template](#) (see page 532)

[How To Protect a Resource with a SAML 1.x Authentication Scheme](#) (see page 295)

## 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

**More Information:**

[Authentication Scheme Dialog--SAML POST Template--Advanced Tab](#) (see page 538)

## Configure SAML 1.x User Disambiguation

When you configure an authentication scheme, you specify how 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 value in the assertion; however, you can also obtain the LoginID by specifying an Xpath query.

After the authentication scheme determines the LoginID, Federation Security Services looks at the search specification configured for the authentication scheme. For SAML 1.x deployments, a search specification is required. The LoginID is not passed back to the Policy Server. Instead, the value of the search specification is used to locate the correct user in the user store.

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 with a search specification.

**Note:** The use of Xpath is optional.

### Obtain the LoginID

You can find the LoginID in two ways:

- Relying on the default behavior, where the LoginID is extracted from the NameID in the assertion. This option requires no configuration.
- Using an Xpath query to find the LoginID in place of the default behavior.

#### To use an Xpath query to determine the LoginID

1. Navigate to the Scheme Setup tab for SAML 1.x artifact or POST authentication template.

2. In the Search Data XPATH field, enter an Xpath query. This query enables the authentication scheme to obtain a LoginID.

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

Xpath queries should not contain namespace prefixes. The following 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()
```

3. Click OK to save your configuration changes.

## Specify a Search Specification to Locate a User (SAML 1.x)

After obtaining the LoginID, configure a search specification to locate the user.

### To locate a user with a search specification

1. From the Scheme Setup dialog, select a namespace and click Edit, to the right of the namespace box.

The SiteMinder Authentication Scheme Namespace Mapping dialog box opens.

In the Search Specification field, enter a user directory attribute followed by %, which represents the LoginID. The authentication scheme uses this attribute to search the namespace.

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.

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

2. Click OK to save your changes.

You return to the Authentication Scheme Properties dialog.

The combination of the LoginID value and the search specification define the disambiguation process.

## Customize Assertion Processing with the Message Consumer Plug-in

The Message Consumer Plug-in is SiteMinder's Java program that implements the Message Consumer Extension API. Using this plug-in you can implement your own business logic for processing assertions, such as rejecting an assertion and returning a SiteMinder-defined status code. This additional processing works together with SiteMinder's 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 is configured. The plug-in then has the opportunity to disambiguate the user, if it knows how.

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 should return a `UserNotFound` error, which is documented in the `MessageConsumerPlugin` interface. The plug-in's use of SiteMinder's redirect URLs feature is optional and is based on the error code returned by the plug-in. If the Message Consumer plug-in is not configured, the redirect URLs are used based on the error generated by the SAML authentication scheme.

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, but this is not required.

To integrate the Message Consumer plug-in with SiteMinder, use the parameter values that you specify for the plug-in configuration. The plug-in configuration is part of the SAML 1.x, SAML 2.0 and WS-Federation authentication scheme configuration.

### More information:

[Specify Redirect URLs for Failed SAML 1.x Authentication](#) (see page 293)

[Specify Redirect URLs for Failed SAML 2.0 Authentication](#) (see page 388)

[Specify Redirect URLs for Failed WS-Federation Authentication](#) (see page 449)

## Configure the SAML 1.x Message Consumer Plug-in

### To configure the message consumer plug-in for artifact or POST authentication

1. From the Authentication Scheme Properties dialog, click Additional Configuration.

The SAML 1.x. Auth Scheme Properties dialog opens.

2. Implement the plug-in class.

A sample class, `MessageConsumerPluginSample.java`, can be found in `sdk/samples/messageconsumerplugin`.

3. In the Full Java Class Name field, enter the Java class name of the plug-in. This plug-in is invoked by the Message Consumer at run time.

The plug-in class can parse and modify the assertion, and then return the result to the Message Consumer for final processing.

Only one plug-in is allowed for each authentication scheme. For example, `com.mycompany.messageconsumer.samplecode`

**Note:** Specify a Message Consumer plug-in for each authentication scheme.

4. Click OK to save the changes.

You return to the Authentication Scheme Properties dialog.

### More Information:

[Integrate the Message Consumer Plug-in for SAML 1.x Authentication](#) (see page 287)

## Integrate the Message Consumer Plug-in for SAML 1.x Authentication

After writing a message consumer plug-in, integrate the plug-in with the SAML 1.x authentication scheme.

The instructions for compiling the message consumer plug-in Java file are in the `MessageConsumerPluginSample.java` file located in `sdk/samples/messageconsumerplugin` as well as the provided build scripts.

### To integrate the message consumer plug-in with the authentication scheme

1. Compile the message consumer plug-in Java file.

The Java file requires the following .jar file installed with the Policy Server:

`policy_server_home/bin/java/SmJavaApi.jar`

2. In the `JVMOptions.txt` file, modify the `-Djava.class.path` value so it is set to the classpath for the plug-in. This enables the plug-in to be loaded.

**Note:** Do not modify the classpath for `SMJavaApi.jar`.

3. In the Policy Server User Interface, specify the plug-in that SiteMinder should use. Click Additional Configuration in the SAML Auth Scheme Properties dialog and complete the following fields:

**Full Java Class Name**

Specify the Java class name for the plug-in. For example, a sample class included with the 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. Restart the Policy Server.

Restarting the Policy Server ensures that the latest version of the assertion plug-in is picked up after being recompiled.

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*.

## Supply SAML Attributes as HTTP Headers

An assertion response may include attributes in the assertion. These attributes can be supplied as HTTP header variables and used by a client application can use these headers for finer grained access control.

The benefits of including attributes in HTTP headers is 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 seen by the user's browser, which reduces security concerns.

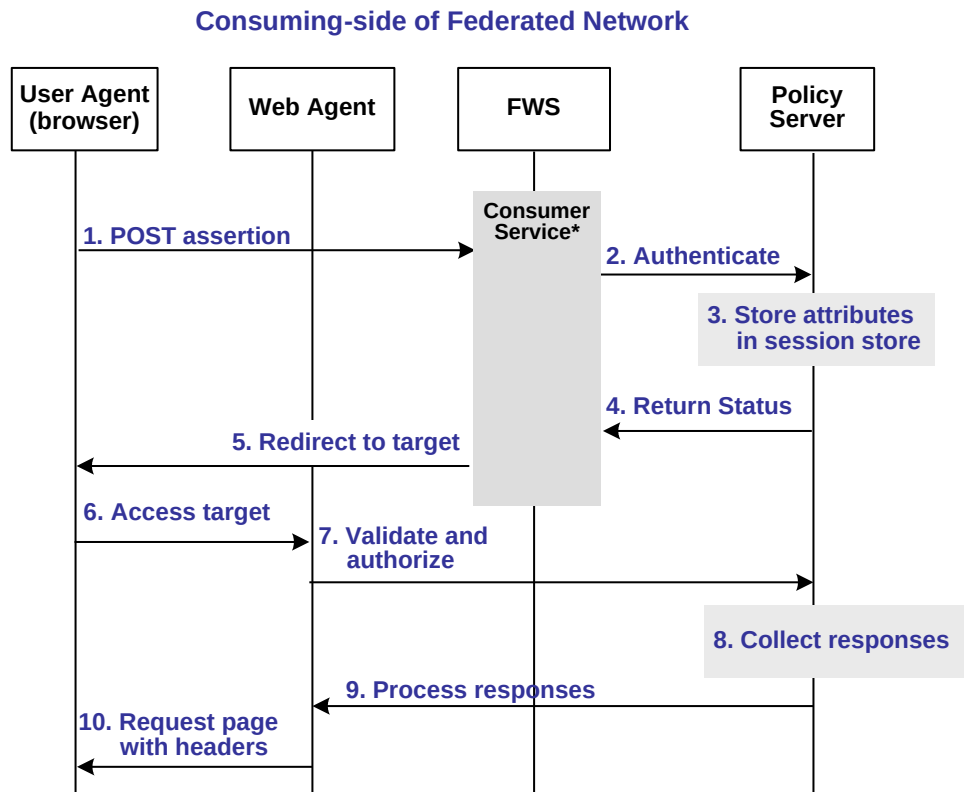
**Note:** The HTTP headers have size restrictions that the attributes cannot exceed. Federation Security Services can send an attribute in a header up to the web server size limit for a header. Only one assertion attribute per header is allowed. See the documentation for your web server to determine 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's application.

The following flow diagram shows the sequence of events at runtime:



\*Consumer service can be one of the following:

- SAML Credential Collector (SAML 1.x)
- Assertion Consumer Service (SAML 2.0)
- Security Token Consumer Service (WS-Federation)

**Note:** The SPS federation gateway can replace the Web Agent and Web Agent Option Pack to provide the SiteMinder 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 *CA SiteMinder 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 producing partner, it sends the assertion to the appropriate consumer service at the consuming partner. 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's 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's session and to ensure 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

There are several configuration steps required to retrieve the SAML attributes cached in the session store and provide them as HTTP headers.

**The components you must configure are as follows**

1. Select PersistAttributes as the redirect mode for the SAML authentication scheme. This enables the SAML Attributes to be returned as HTTP headers.
2. Configure an authorization rule for the realm that contains the target resource.
3. Configure a response that uses the active response type for each SAML attribute to be supplied as a header.
4. 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 Policy Server User Interface.
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, you need to 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, therefore the Web Agent cannot re-authenticate the user and pass on the HTTP headers. So, the retrieval of the attributes must happen during the authorization stage.

### To create an OnAccessAccept Rule for the realm

1. Log on to the Policy Server User Interface.
2. From the Domains tab, navigate to the realm which protects the target resource.
3. Select the realm with the target resource and choose 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. Choose 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 group box..

8. Ensure that Enabled is checked in the Allow/Deny and Enable/Disable group box.
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

You must configure a response that sends the SAML attributes as HTTP headers to the Web Agent. The Web Agent will process the response and make the header variables available to the client application.

### To create a response to send the attributes as headers

1. Log on to the Policy Server User Interface.
2. From the Domains tab, select the domain that contains the target resource.
3. In the domain, select the Responses object and create a new 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 group box.
7. Complete the fields in the Attribute Fields group box as follows:

#### Variable Name

Specify the name you want for the header variable. You assign this name.

#### Library Name

smfedattrresponse

This must be the entry for this field.

#### Function Name

getAttributeValue

This 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 will be in the assertion.

8. Click on OK to save the attribute.
9. Repeat the procedure for each attribute that should become an HTTP header variable. You can configure many attributes for a single response.

The response will send 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 Policy Server User Interface.
2. From the Domains tab, select the domain that contains the target resource.
3. In the domain, select the Policies object and create a new policy.  
The Policy Properties dialog opens.
4. Enter a descriptive name in the Name field.
5. Select the users that should 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.

## Specify Redirect URLs for Failed SAML 1.x Authentication

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.

### More information:

[Authentication Scheme Dialog--SAML 1.x Auth Scheme Properties \(Artifact, POST\)](#) (see page 538)

## How To Protect a Resource with a SAML 1.x Authentication Scheme

At the consumer, you must configure a SAML 1.x artifact or POST profile authentication scheme for each producer that generates assertions. After that authentication scheme is created, you can use it to protect federation resources.

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 being requested by users.

There are two ways to set-up a realm that includes a SAML authentication scheme:

- You can create a unique realm for each authentication scheme already configured.
- You can configure a single target realm 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. After configuring a realm, 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. An intersite transfer URL redirects a user from the producer to the consumer, and the target URL is specified in the URL's TARGET variable. At the producer site, an administrator needs to include this URL in a link so that this link the user gets redirected to the consumer.

## How To Assign a SAML 1.x Authentication Scheme to a Realm

There are two ways to set-up a realm that includes a SAML authentication scheme:

- You can create a unique realm for each authentication scheme/producer already configured.
- You can configure a single target realm 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.

**Important!** Each target URL in the realm is also identified in an intersite transfer URL. An intersite transfer URL redirects a user from the producer to the consumer, and the target URL is specified in the URL's TARGET variable. At the producer site, an administrator needs to include this URL in a link so that this link the user gets redirected to the consumer.

## Configure a Unique Realm for Each SAML 1.x Authentication Scheme

The process for configuring a unique realm for each SAML authentication scheme (artifact or profile) follows the standard instructions for creating realms in the Policy Server User Interface.

### To create a realm for each SAML authentication scheme

1. Create a policy domain that will contain the realm with the target resources.
2. Create a realm in the 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.
3. Create a rule for the realm.
4. Include the realm and associated rule in a policy that protects the target resource.

The resources are now protected by a SAML authentication scheme.

## Configure a Single Target Realm for All SAML 1.x Auth. Schemes

To simplify configuration of realms for SAML authentication schemes, you can create a single target realm for multiple producers.

To do this, set-up:

- A single custom authentication scheme  
You should have already configured SAML artifact and/or SAML POST profile authentication schemes for each producer.
- A single realm with one target URL

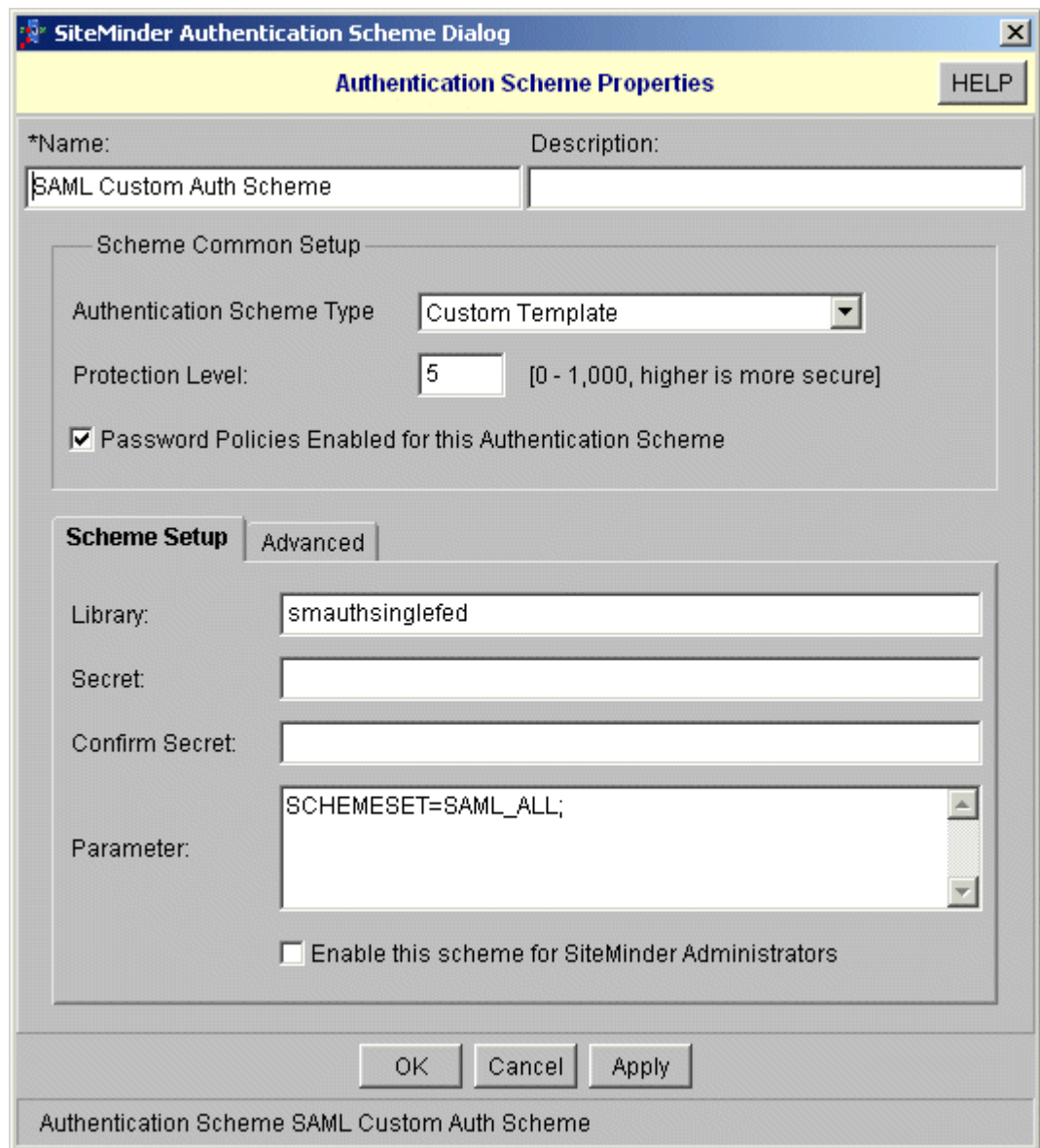
### To configure a custom authentication scheme for the single realm:

1. Log into the Policy Server User Interface.
2. From the menu bar, select Edit, System Configuration, Create Authentication Scheme.

The Authentication Scheme Properties dialog box opens.



3. From the Authentication Scheme Type drop-down list, select Custom Template. The contents of the dialog box change for the custom template.



The image shows a screenshot of the 'SiteMinder Authentication Scheme Dialog' window, specifically the 'Authentication Scheme Properties' tab. The window has a title bar with 'SiteMinder Authentication Scheme Dialog' and a close button. Below the title bar is a yellow header with 'Authentication Scheme Properties' and a 'HELP' button. The main area is divided into sections. The first section has two labels: '\*Name:' and 'Description:'. The '\*Name:' field contains 'SAML Custom Auth Scheme'. Below this is a section titled 'Scheme Common Setup' which contains a dropdown for 'Authentication Scheme Type' set to 'Custom Template', a 'Protection Level' field set to '5' with a note '[0 - 1,000, higher is more secure]', and a checked checkbox for 'Password Policies Enabled for this Authentication Scheme'. Below this is a section titled 'Scheme Setup' with two sub-tabs: 'Scheme Setup' (selected) and 'Advanced'. The 'Scheme Setup' sub-tab contains fields for 'Library:' (set to 'smauthsinglefed'), 'Secret:', 'Confirm Secret:', and 'Parameter:' (containing 'SCHEMESET=SAML\_ALL;'). There is also an unchecked checkbox for 'Enable this scheme for SiteMinder Administrators'. At the bottom are 'OK', 'Cancel', and 'Apply' buttons. A status bar at the very bottom reads 'Authentication Scheme SAML Custom Auth Scheme'.

**SiteMinder Authentication Scheme Dialog**

**Authentication Scheme Properties** HELP

\*Name:  Description:

**Scheme Common Setup**

Authentication Scheme Type:

Protection Level:  [0 - 1,000, higher is more secure]

☒ Password Policies Enabled for this Authentication Scheme

**Scheme Setup** Advanced

Library:

Secret:

Confirm Secret:

Parameter:

☐ Enable this scheme for SiteMinder Administrators

OK Cancel Apply

Authentication Scheme SAML Custom Auth Scheme

4. In the Library field, enter smauthsinglefed for the library name.
5. Disregard the Secret and Confirm Secret fields.

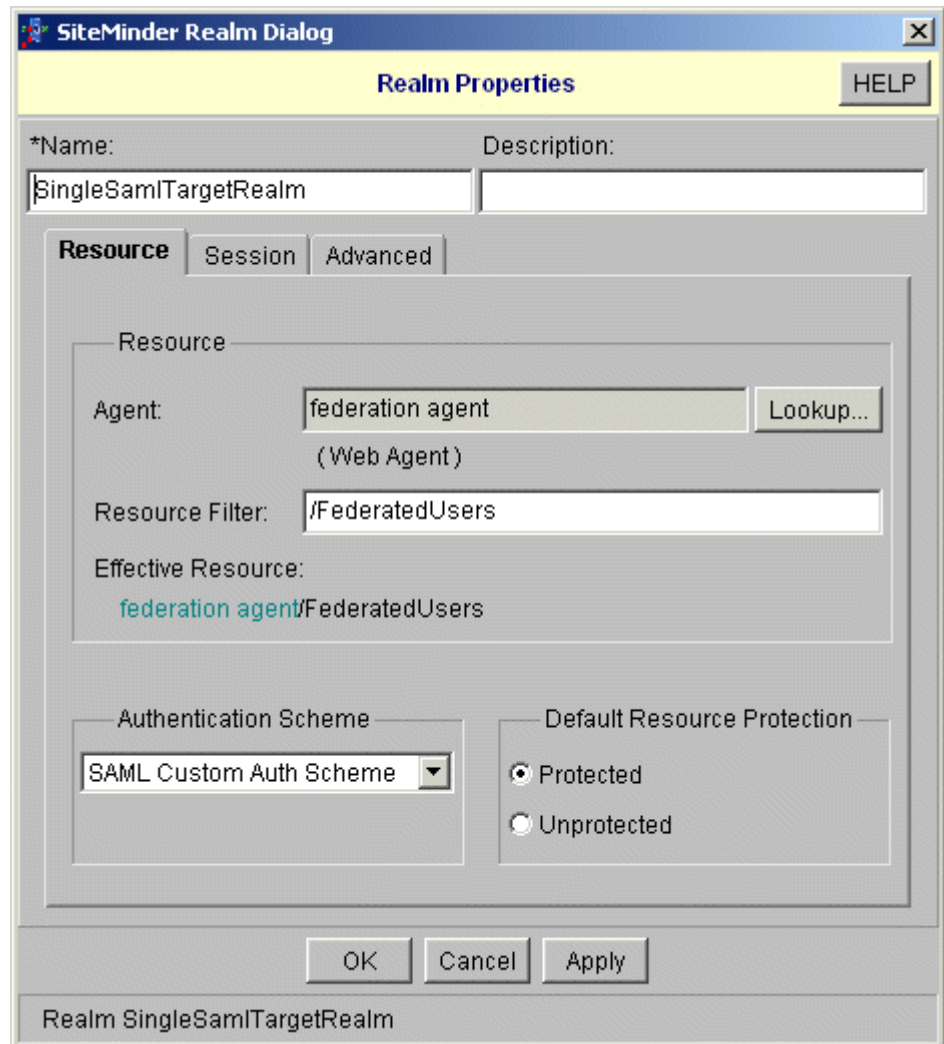
6. In the Parameter field, specify one of the following:
  - **SCHEMESET=LIST; <saml-scheme1>;<saml\_scheme2>**  
Specifies the list of SAML authentication scheme names to use. If you configured an artifact scheme called artifact\_producer1 and POST profile scheme called samlpost\_producer2, you will enter these schemes.
  - **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.
7. Disregard the Enable this scheme for SiteMinder Administrators checkbox.
8. Click OK to save your changes.

**To create the single target realm:**

1. Display the list of policy domains.
2. Expand the policy domain where you will add the realm.
3. Click on the realm icon.
4. From the menu bar, select Edit, Create Realm.  
The SiteMinder Realm dialog box opens.
5. In the Name field, enter a name for this custom target realm.
6. In the Agent field, select a SiteMinder Web Agent protecting the Web server with the target consumer resource.
7. In the Resource Filter field, specify the location of the target resource to which any user from any producer site should be redirected.  
For example, /FederatedUsers.
8. From the Authentication Scheme drop-down list, select the custom authentication scheme that you configured for directing requests to the appropriate SAML authentication schemes.

9. Ensure Protected is selected in the Default Resource Protected group box.
10. Click OK to save the realm configuration.

The following graphic shows the Realm Properties dialog for the custom realm.



## Create a Rule for the Realm

After you create a realm containing target resources, you associate a rule with the realm. Part of the rule configuration is to assign a rule action, which allows you to control processing that occurs when users authenticate to gain access to a resource.

The rule actions you can configure for SAML 1.x are:

- Web Agent action (Get, Post, or Put)
- Authentication event with the OnAuthReject action

To create this rule select the Authentication events radio button along with the OnAuthReject action from the pull-down menu.

If you want to use this type of rule action you must also configure a Web Agent action.

## Create a Response Associated with the Rule

If you configure a rule triggered by an Authentication event with the OnAuthReject action, you must also configure a Web-Agent-OnAuth-Reject-Redirect response, in which the response variable is the URL where the user is redirected if he fails to authenticate. The rule and response grouped together in a policy enables you to redirect a user to an error page or a login page, for example.

## Form the Policy to Protect the Target Resource

The general process for creating the policy is as follows:

**Note:** This assumes that a user directory has already been configured.

1. Create a SAML 1.x authentication scheme.
2. Associate the SAML authentication scheme with a realm.
3. Within the realm, create one rule with a Web Agent action or two rules, one with a Web Agent action and the other as an authenticating event with the OnAuthReject action.
4. Optionally, if you configured an OnAuthReject rule, configure a Web-Agent-OnReject-Redirect Response attribute that contains a value of a URL where the rejected user should be redirected. We recommend using a static attribute value.
5. Create a policy grouping the realm, rule, and response together.
6. Exit the Policy Server User Interface.

# Chapter 12: Configure SiteMinder as a SAML 2.0 Identity Provider

---

This section contains the following topics:

- [Configuration Checklist at the Identity Provider](#) (see page 301)
- [Add a SAML 2.0 Service Provider to an Affiliate Domain](#) (see page 303)
- [Select Users For Which Assertions Will Be Generated](#) (see page 303)
- [Specify Name Identifiers for SAML 2.0 Assertions](#) (see page 306)
- [Configure Required General Information](#) (see page 308)
- [Configure Single Sign-on for SAML 2.0](#) (see page 311)
- [Allow Access to the Federation Web Services Application](#) (see page 322)
- [Set Up Links at the IdP or SP to Initiate Single Sign-on](#) (see page 323)
- [Configure Attributes for Inclusion in Assertions \(optional\)](#) (see page 329)
- [Configure Single Logout \(optional\)](#) (see page 333)
- [Enable Identity Provider Discovery Profile \(optional\)](#) (see page 335)
- [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 340)
- [Protect the Authentication URL to Create a SiteMinder Session \(SAML 2.0\)](#) (see page 343)
- [Protect the Artifact Resolution Service with Client Certificate Authentication \(optional\)](#) (see page 345)

## 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](#) (see page 302)
- [Optional Configuration Tasks](#) (see page 302)

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 503).
- 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 623).

## Required Configuration Tasks to Identify a Service Provider

The tasks required to identify a Service Provider are as follows:

- Create an affiliate domain.
- Add a Service Provider to the affiliate domain and:
  - Specify the name of the Service Provider.
  - Specify and protect the Authentication URL.
  - Select users from a user store for which assertions will be generated.
  - Specify the Name ID on the Name IDs tab.
  - Specify the SP ID and the IdP ID on the General tab.
  - Complete the Audience and Assertion Consumer Service fields on the SSO tab.
  - 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 configuring SSO.

## Optional Configuration Tasks for Identifying a Service Provider

The following are optional tasks for identifying a Service Provider:

- Configure single sign-on restrictions:
  - Set IP address restrictions to limit the addresses used to access Service Providers.
  - Configure time restrictions for Service Provider operations.
- Enable enhanced client or proxy profile.
- Configure attributes for inclusion in assertions.
- Configure single logout (SLO).
- Configure the Identity Provider Discovery profile.
- Encrypt the Name ID 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 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 Policy Server User Interface.
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:

[Protect the Authentication URL to Create a SiteMinder Session \(SAML 2.0\)](#) (see page 343)

[SAML Service Provider Properties Dialog Fields and Controls](#) (see page 542)

[Integrate the Assertion Generator Plug-in with SiteMinder \(SAML 2.0/WS-Federation\)](#) (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 Policy Server User Interface.
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 Policy Server User Interface 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.

**More Information:**

[SAML Service Provider Properties Dialog Fields and Controls](#) (see page 542)

## 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 Policy Server User Interface 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 Policy Server User Interface 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 319)

## Configure a Name ID

### To configure a name ID

1. Log in to the Policy Server User Interface 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.

**More Information:**

[SAML Service Provider Dialog--Name IDs Tab](#) (see page 545)

## 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.

**More Information:**

[SAML Affiliations Reference](#) (see page 583)

## 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 Policy Server User Interface.
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 252) 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 334) is calculated to understand more about the skew time.

### **More Information:**

[SAML Service Provider Dialog--General Tab](#) (see page 547)

## 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.

## 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 Policy Server User Interface.

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 Policy Server User Interface 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 Policy Server User Interface, 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.

**More Information:**

[Manage the Key Database for Signing and Encryption](#) (see page 479)

## 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 Policy Server User Interface provide guidance for configuring various settings.

### To configure single sign-on at the Identity Provider

1. Log on to the Policy Server User Interface.
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](#) (see page 541) 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.

### More information

[SAML Service Provider Dialog--SSO Tab](#) (see page 550)

## Assertion Validity for Single Sign-on

For single sign-on, the values of the Skew Time and the Validity Duration determine how SiteMinder FSS calculates the total time that an assertion is valid. FSS 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 the NotBefore and NotOnOrAfter values represent the beginning and end of the validity interval.

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

**NotBefore=IssueInstant - Skew Time**

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

**Calculating Assertion Validity if FSS is at Both Sides of the Partnership**

If FSS is at both sides of a partnership, the total time the assertion is valid 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 FSS, 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 takes the NotBefore and NotOnOrAfter values from the assertion and applies its skew time to those values to calculate new NotBefore and NotOnOrAfter values.

Skew Time = 180 seconds (3 minutes)

NotBefore = 4:56PM

NotOnOrAfter=5:05PM

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

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

## 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 to determine if a user session becomes invalid and the user has 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 Policy Server User Interface.
2. Select the Service Provider entry 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 you enter is the value of the SessionNotOnOrAfter parameter in the assertion.

The options are:

**Use Assertion Validity**

Calculates the SessionNotOnOrAfter value 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 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 SAML\_SP\_CUSTOM\_TIME\_OUT property.

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

6. Click OK to save the changes.

## Define 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 Policy Server User Interface.
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.

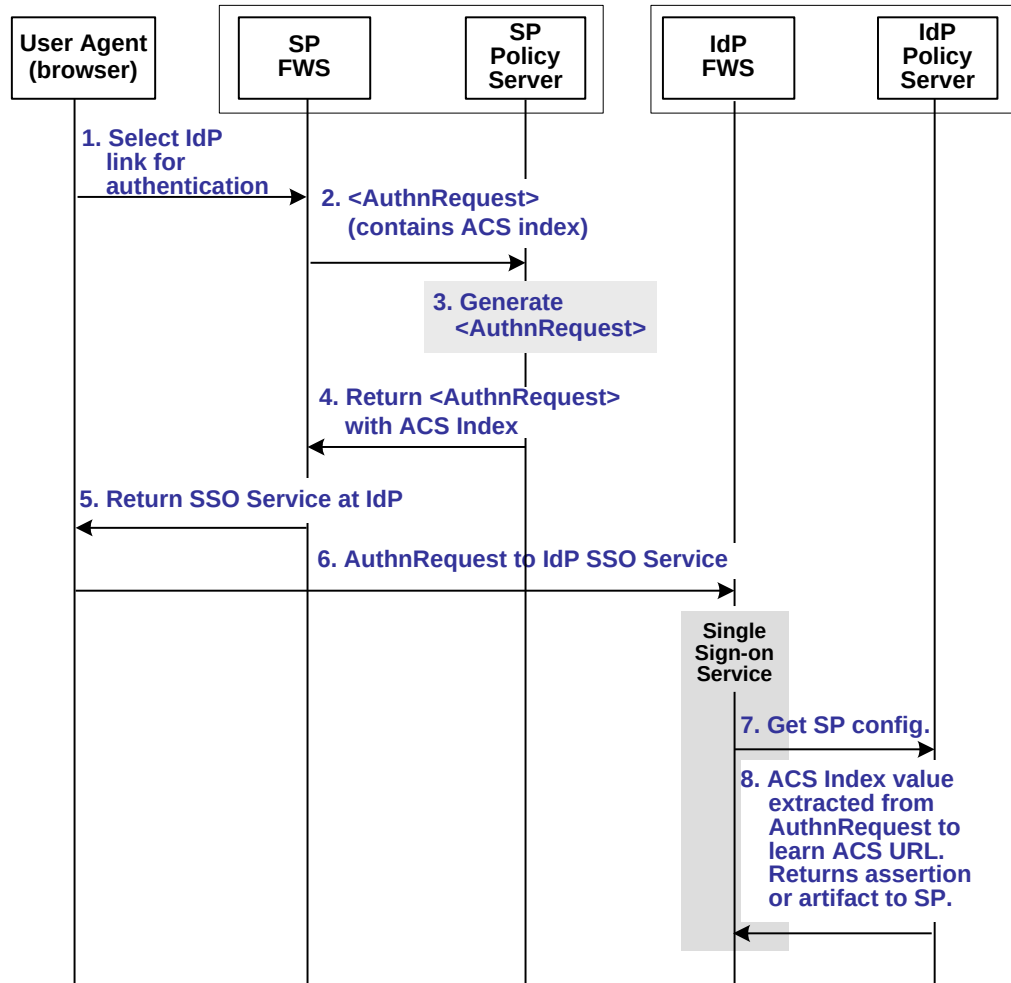
**Note:** Remember to configure index entries in the SAML 2.0 authentication scheme at the Service Provider.

### More Information:

[Configure Single Sign-on at the SP](#) (see page 364)  
[Assertion Consumer Services Dialog](#) (see page 553)

## Indexed Endpoints Flow Diagram

The following diagram 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. In the request it sends, it includes the IdP ID and optionally, the AssertionConsumerServiceIndex and ForceAuthn query parameters.

**Note:** 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 in the AuthnRequest, 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 checked against the IdP metadata collected from an IdP descriptor document 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 checked to see if the AssertionConsumerServiceIndex corresponds 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 stating 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 AuthnRequest is then redirected by the SP FWS application 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 based on the AssertionConsumerServiceIndex, unless the value of the Index is not found in the SP metadata. If the Index is not found, an error is generated stating that an invalid AssertionConsumerServiceIndex is specified in the AuthnRequest message.

The Assertion Consumer URL is used by the single sign-on service to send the assertion or artifact to the SP, depending on the binding associated with the SP.

**Note:** If the AssertionConsumerServiceIndex parameter is not specified in the incoming AuthnRequest, the value of the Assertion Consumer Service and the corresponding binding are used by default.

**More Information:**

[SAML Service Provider Dialog--SSO Tab](#) (see page 550)

## Define Indexed Endpoints for Different Single Sign-on Bindings

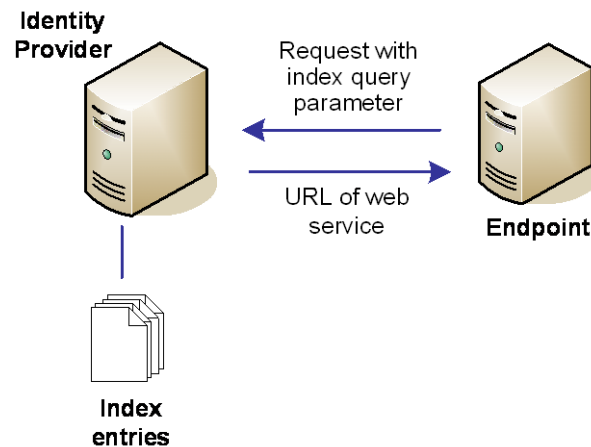
Federation Security Services offers you the ability to configure indexed endpoints. An indexed endpoint is the site where assertions are consumed. In the context of SiteMinder, this endpoint is the Service Provider where the Assertion Consumer Service resides.

Each endpoint you configure is assigned a unique index value, instead of an single, explicit reference to an Assertion Consumer Service URL. The assigned index is added to a Service Provider's request for an assertion that it 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 may also want to configure different protocol bindings (artifact vs. POST) for the Assertion Consumer Service by assigning more than one endpoint to the service.

**Note:** If your federated network contains a mix of SiteMinder versions, for example, the Service Provider is version 6.0 SP 5 and the Identity Provider is version 6.0 SP 6, you cannot configure indexed endpoints. Simply configure only one Assertion Consumer Service for both HTTP bindings.

The following figure shows a network that benefits from indexed endpoints.



## 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 level you configure in the Authentication Level field on the SAML Service Provider Properties dialog.

If the protection level for the Authentication URL is less than the level set in the Authentication Level field, SiteMinder will not generate an assertion.

## 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 Policy Server User Interface.
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 Policy Server User Interface 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 Policy Server User Interface 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.

**More Information:**

[Restrictions Dialog for an SP](#) (see page 553)

## 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 Policy Server User Interface 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.

**More Information:**

[Restrictions Dialog for an SP](#) (see page 553)

## Allow Access to the Federation Web Services Application

After you add affiliates to an affiliate domain, the affiliates need permission to access the Federation Web Services application. When you install the Policy Server and Policy Server Option Pack, the FederationWebServicesDomain is installed by default. This domain includes the following policies:

- FederationWSAssertionRetrieval
- FederationWSNotificationService
- FederationWSSessionServicePolicy
- SAML2FWSArtifactResolutionServicePolicy

### To specify permission to the Federation Web Services application

1. From the Domains tab, expand FederationWebServicesDomain and select Policies.
2. Select one of the policies, and click Edit, Properties of Policy.

For SAML 1.x, you need to permit access to:

- FederationWSAssertionRetrieval
- FederationWSNotificationService
- FederationWSSessionServicePolicy

For SAML 2.0, you need to permit access to  
SAML2FWSArtifactResolutionServicePolicy

The SiteMinder Policy dialog opens.

3. From the Users tab, select one of the following:
  - FederationWSCustomUserStore tab for SAML 1.x
  - SAML2FederationCustomUserStore tab for SAML 2.0.

The Users/Groups dialog opens.

The consumers, Service Providers, and Resource Partners are the "users" included in the listed user stores.

4. Click Add/Remove on the appropriate tab.
5. From the Available Members list, choose the affiliate domains that should have access to Federation Web Services then move them to the Current Members list.
6. Click OK to return to the Policy List.
7. Repeat this procedure for all policies relevant for the SAML version you are using.

## 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, an unsolicited response at the Identity Provider needs to be initiated. To initiate an unsolicited response, you need to create a hard-coded link that generates an HTTP Get request that is accepted by the Federation Web Service application and the Assertion Generator. This HTTP Get request must contain a query parameter that provides the Service Provider ID for which the Identity Provider needs to generate the SAML assertion response. A user clicks this link 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. This URI is defined by the SAML 2.0 specification.

The binding must also be specified in the SAML Service Provider properties for the unsolicited response to work.

Note the following:

- If you do not include the ProtocolBinding query in the link, and only one binding is configured in the Service Provider properties, that binding is used.
- If artifact and POST are enabled in the Service Provider properties POST is the default. Therefore, if you want to *only* use artifact binding, you must include the ProtocolBinding query parameter in the link.

**Important!** If you configure indexed endpoint support for Assertion Consumer Services, the binding you choose for the Assertion Consumer Service is overridden by the value of the ProtocolBinding query parameter in the link for an unsolicited response.

**More information:**

[Unsolicited Response Query Parameters Used by a SiteMinder IdP](#) (see page 324)

## 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 will fail.

#### Required Use of the ProtocolBinding Query Parameter

Use of the ProtocolBinding query parameter is required *only* if artifact and POST binding are enabled for the Service Provider properties and the user wants to only use artifact binding.

- The URI for the artifact binding, as specified by the SAML 2.0 specification is:  
urn:oasis:names:tc:SAML:2.0:bindings:HTTP-Artifact
- The URI for the POST binding, as specified by the SAML 2.0 specification is:  
urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST

You do not need to set this parameter for HTTP-POST single sign-on.

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

#### Example: Unsolicited Response with ProtocolBinding

This link redirects the user to the Single Sign-on service. Included in this link is the Service Provider identity, specified by the SPID query parameter and the artifact binding is being used, as specified by the bindings query parameter. After the user clicks this hard coded link, they are redirected to the local Single Sign-on service.

```
http://idp-ca:82/affwebservices/public/saml2sso?SPID=http%3A%2F%2Ffedsrv.
acme.com
%2Fsmidp2for90&ProtocolBinding=urn:oasis:names:tc:SAML:2.0:bindings:HTTP-
Artifact
```

### Optional Use of the ProtocolBinding Query Parameter

When you *do not* use the ProtocolBinding query parameter the following applies:

- If only one binding is enabled for the Service Provider and the ProtocolBinding is not specified in the unsolicited response, the enabled binding is used.
- If both bindings are enabled for the Service Provider and the ProtocolBinding is not specified in the unsolicited response, the POST binding is used 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, specified by the SPID query parameter. There is no ProtocolBinding query parameter. 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. You can use the RelayState query parameter to indicate the target destination; however, this method is optional because there may be a configuration mechanism at the Service Provider itself to indicate the target.

You should 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)

If a user visits the Service Provider first and then goes to an Identity Provider, you have to create an HTML page at the Service Provider containing hard-coded links to the AuthnRequest service at the Service Provider. These links redirect the user to the Identity Provider to be authenticated as well as determining what is included in the AuthnRequest itself.

The hard-coded link that the user selects must contain specific query parameters. These parameters are supported by an HTTP GET request to the AuthnRequest service at the Service Provider's Policy Server.

**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 that is hosting the Web Agent Option Pack or the SPS federation gateway.

**IdP\_ID**

Specifies the identity assigned to the Identity Provider

**URI\_for\_binding**

Identifies the URI of the POST or Artifact binding for the ProtocolBinding element. This URI is defined by the SAML 2.0 specification.

A binding must also be enabled for the SAML authentication scheme for the request to work.

Note the following:

- If you do not include the ProtocolBinding query parameter in the AuthnRequest link, the default binding used 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 defined at the Identity Provider is used.
- If artifact and POST are enabled for the SAML authentication scheme, but you only want to use artifact binding, you must include the ProtocolBinding query parameter in the link..

## AuthnRequest Query Parameters Used by a SiteMinder SP

The query parameters a SiteMinder Service Provider can use in the links to the AuthnRequest Service are as follows:

**ProviderID (required)**

ID of the Identity Provider where the AuthnRequest message is sent by the AuthnRequest Service.

**ProtocolBinding**

Specifies the ProtocolBinding element in the AuthnRequest message. This element specifies the protocol used to return the SAML response from the Identity Provider. If the specified Identity Provider is not configured to support the specified protocol binding, the request will fail.

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

Use of the ProtocolBinding parameter is required if artifact and POST binding are enabled for an authentication scheme and the user wants to use only the artifact binding.

- The URI for the artifact binding, as specified by the SAML 2.0 specification is:  
urn:oasis:names:tc:SAML:2.0:bindings:HTTP-Artifact
- The URI for the POST binding as specified by the SAML 2.0 specification is:  
urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST

You do not need to set this parameter for HTTP-POST single sign-on.

#### 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
```

After a user clicks the link at the Service Provider, the Federation Web Services application passes a request for an AuthnRequest message from the local Policy Server.

### Optional Use of ProtocolBinding

When you *do not* use the ProtocolBinding query parameter the following applies:

- If only one binding is enabled for the authentication scheme and the ProtocolBinding query parameter is not specified, the enabled binding for the authentication scheme is used.
- If both bindings are enabled and the ProtocolBinding query parameter is not specified, POST binding is used as the default.

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

#### Example: AuthnRequest Link without ProtocolBinding

This sample link goes to the AuthnRequest service. It specifies the Identity Provider in the ProviderID query parameter.

```
http://ca.sp.com:90/affwebservices/public/saml2authnrequest?ProviderID=
http%3A%2F%2Ffedsrv.acme.com%2Fsmidp2for90
```

After a user clicks the link at the Service Provider, the Federation Web Services application passes a request 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 the SP sets ForceAuthn=True in the AuthnRequest message, and a SiteMinder session exists for a particular user, the Identity Provider re-challenges the user for credentials. If the user successfully authenticates, the IdP sends the identity information from the existing session in the assertion and discards the session generated for the re-authentication.
- 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 in the SAML 2.0 authentication scheme configured using the Policy Server User Interface. The authentication scheme also has an option to override the target with the RelayState query parameter if you choose.

You should 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 or not 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 may 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. It 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 single sign-on is initiated by a Service Provider, that Service Provider may include a ForceAuthn or IsPassive query parameter in an AuthnRequest message.

When a Service Provider includes ForceAuthn or IsPassive in the AuthnRequest, a SiteMinder Identity Provider handles these query parameters as follows:

### ForceAuthn Handling

When a Service Provider includes ForceAuthn=True in the AuthnRequest, a SiteMinder Identity Provider does the following:

- If ForceAuthn=True in the AuthnRequest message, and a SiteMinder session exists for a particular user, the SiteMinder Identity Provider re-challenges the user for credentials. If the user successfully authenticates, the IdP sends the identity information from the existing session in the assertion and discards the session generated for the re-authentication.
- If 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 it cannot be honored by the Identity Provider, one of the following SAML responses is sent back to the Service Provider:

- If IsPassive=True in the AuthnRequest message and there is no SiteMinder session, a SiteMinder Identity Provider returns a SAML response that includes an error message because SiteMinder requires a session.
- If IsPassive=True in the AuthnRequest message and there is a SiteMinder session, the SiteMinder Identity Provider returns the assertion.
- If IsPassive and ForceAuthn are in the AuthnRequest message and both are set to True, the SiteMinder Identity Provider returns an error because this is an invalid request. IsPassive and ForceAuthn are mutually exclusive.

## Configure Attributes for Inclusion in 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:** Attributes statements are not required in an assertion.

Attributes are used by servlets, Web applications, or other custom applications 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 and set it to a maximum dollar amount 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, Federation Security Services 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, Federation Security Services 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.

You 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 that Function for SSO and Attribute Query Requests

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 attributes function is determined by the Retrieval Method field in the SAML Service Provider Attribute dialog.

If you want the same attribute to be used for both services, you must 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.

## 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.

### 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.

### More Information:

[SAML Service Provider Dialog--Attributes Tab](#) (see page 555)

## 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.

## 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 Policy Server User Interface.

### To configure single logout

1. Log in to the Policy Server User Interface 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.

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 334)

## Single Logout Request Validity

The values of the SLO validity duration (Validity Duration field on the SLO tab) 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.

To determine when the logout request is no longer valid, the Policy Server takes its current system time and adds the Skew Time plus the SLO Validity Duration together. 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, you should 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 should be an unprotected local resource at the Service Provider site.
- If single logout is initiated at an Identity Provider site, the logout confirmation page should 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 should also support the following:

- Be able to handle Base 64-encoded data and read cookies.
- The page at the IdP and SP should contain code that looks for a SIGNOUTFAILURE cookie. This cookie is set in the user's browser if single logout fails, and it contains the Partner IDs of the federation sites where logout failed. These IDs are base 64-encoded and 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, which 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 should inform users to close the web browser to remove all session data.

## 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 Policy Server User Interface.
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`

## Encrypt a NameID and an Assertion

You can encrypt the Name ID in an assertion and/or the assertion itself. Encryption adds another level of protection when transmitting the assertion.

When you configure encryption, you must specify the partner certificate, which is included 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 have enabled encryption, when the first federation call is made, the memory of the Policy Server may increase substantially to load the encryption libraries and allocate additional memory.

### Enabling Encryption

#### To implement encryption

1. Log in to the Policy Server User Interface 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.

**More Information:**

[SAML Service Provider Dialog--Encryption Tab](#) (see page 560)

[Manage the Key Database for Signing and Encryption](#) (see page 479)

## Request Processing with a Proxy Server at the IdP

When SiteMinder receives certain requests at the IdP, it validates the message attributes using the local URL for Federation Web Services application before processing the request.

For example, an AuthnRequest message from an SP may 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.

When the SiteMinder federated environment sits behind a proxy server, the local and destination attribute URLs are not the same because the Destination attribute is the URL of the proxy server. For example, the AuthnRequest may include the following Destination attribute:

```
Destination="http://proxy.domain.com:9090/affwebservices/public/saml2sso"
```

The local URL for Federation Web Services, `http://idp.domain.com:8080/affwebservices/public/saml2sso`, does not match the Destination attribute so the 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. 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.

**More Information:**

[SAML Service Provider Dialog--Advanced Tab](#) (see page 561)

## 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 Policy Server User Interface.
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*.

## Integrate the Assertion Generator Plug-in with SiteMinder (SAML 2.0/WS-Federation)

If you write an assertion generator plug-in, you have to integrate the plug-in to work with SiteMinder.

To compile the assertion plug-in Java file, see the instructions in the `SAML2AssertionSample.java` file in the directory:

`sdk/samples/assertiongeneratorplugin`

### To integrate the assertion generator plug-in with SiteMinder

1. Compile the assertion plug-in Java file.

This file requires the following .jar files 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 enables the plug-in to be loaded.

**Note:** Do not modify the classpath for `xercesImpl.jar`, `xalan.jar`, or `SMJavaApi.jar`.

3. In the Policy Server User Interface, specify the plug-in that SiteMinder should use. Access the Advanced tab in the Service Provider Properties or Resource Partner Properties dialog and complete the following fields:

#### Full Java Class Name

Specify a Java class name for an existing plug-in

#### Parameter

Specify a string of parameters to be 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 via the Policy Server User Interface, you can use the Policy Management API (C or Perl) to integrate the plug-in. For instructions, 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.

**To enable the Assertion Generator to include attributes from a web application in an assertion**

1. Compile the assertion plug-in Java file.

This file requires the following .jar files installed with the Policy Server:

- *policy\_server\_home/bin/java/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 enables the plug-in to be loaded.

**Note:** Do not modify the classpath for xercesImpl.jar, xalan.jar, or SMJavaApi.jar.

3. Configure a sample plug-in.

There is an APIContext class in the SMJavaAPI that has a new method, getAttrMap(), which returns a map object containing the attributes from the web application to be included in the assertion. In the SiteMinder SDK, there are two sample Assertion Generator plug-ins that show how to use this map object:

- SAML2AppAttrPlugin.java (SAML 2.0)
- WSFedAppAttrPlugin.java (WS-Federation)

These samples are located in the directory sdk/samples/assertiongeneratorplugin. They enable the Assertion Generator to add attributes from a web application to the Assertion Generator for inclusion in an assertion.

4. In the Policy Server User Interface, specify the plug-in you are using. Access the Advanced tab in the Service Provider Properties or Resource Partner Properties dialog and complete the following fields:

**Full Java Class Name**

Specify the Java class name for the plugin, For example, the sample classes included with the SiteMinder SDK are:

- com.ca.assertiongenerator.SAML2AppAttrPlugin  
(SAML 2.0)
- com.ca.assertiongenerator.WSFedAppAttrPlugin  
(WS-Federation)

### Parameter

Specify a string of parameters to be passed to the plug-in specified in the Full Java Class Name field. These parameters would be the attributes you want to include in the assertion.

**Note:** Instead of specifying the assertion plug-in class and its parameters via the Policy Server User Interface, you can use the Policy Management API (C or Perl) to integrate the plug-in. For instructions, see the *SiteMinder SiteMinder Programming Guide for C* or the *SiteMinder Programming Guide for Java*.

5. Restart the Policy Server.

Restarting the Policy Server ensures that the latest version of the assertion plug-in is picked up after being recompiled.

## Protect the Authentication URL to Create a 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

**To create a policy to protect the Authentication URL**

1. Log into the Policy Server User Interface.
2. From the System tab, create Web Agents to bind to the realms that you will define for the web server at the IdP. You can assign unique Agent names for the web server at the Identity Provider 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 Service Provider resources.
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 Web Server at the Identity Provider.

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 references the following:

- `web_agent_home/affwebservices/redirectjsp`
  - `sps_home/secure-proxy/Tomcat/webapps/affwebservices/redirectjsp`
- c. For the remaining settings, accept the defaults or modify as needed.



6. For HTTP artifact binding only, select the Session tab and check the Persistent Session check box.

To enable single sign-on using the SAML artifact binding, configure a persistent session for the Identity Provider realm. If you do not configure a persistent session, the user cannot access Service Provider 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. Select the Web Agent actions GET, POST, and PUT as the allowed actions.
9. Create a policy for the Web Server at the Identity Provider that includes the rule created in the previous step.

## Protect the Artifact Resolution Service with Client Certificate Authentication (optional)

For SAML 2.0 HTTP-artifact SSO, the Artifact Resolution Service retrieves the assertion stored in the SiteMinder session server at the Identity Provider so it can be sent to the Service Provider. This service needs to be protected with a SiteMinder policy so assertions are retrieved only by authorized users.

By default, there is a pre-configured policy that uses the Basic over SSL authentication scheme to protect the Artifact Resolution Service. When you configure a policy that uses the client certificate authentication scheme to protect this service, this policy must be created for a different realm than the realm that uses the Basic over SSL scheme.

Generally, the administrator at the Identity Provider should create two policies to protect the Artifact Resolution Service. One policy that uses Basic over SSL and the other that uses client certificate authentication.

To protect the Artifact Resolution Service with a client certificate authentication scheme, you:

- Create a policy at the Identity Provider that uses an X.509 client certificate authentication scheme.
- Enable client certificate authentication at the consumer.

### Using Client Certificate Authentication with an IIS 5.0 Web Server

Client certificate authentication is not supported for IIS 5.0 Web servers at the producer/Identity Provider. However, it can be used on an IIS 5.0 Web server at the consumer/Service Provider to communicate with a non-SiteMinder producer/Identity Provider.

To work around this issue, use the IIS 5.0 Web server's client certificate functionality at the producer/Identity Provider and do not configure SiteMinder's client certificate functionality. If you apply this work around, be aware that the CN portion of the certificate's DN value must contain the affiliate name value.

#### More Information:

[Create the Artifact Resolution Service Policy](#) (see page 346)

## Create the Artifact Resolution Service Policy

### To create a policy for the Artifact Resolution Service

1. For each Service Provider, add an entry to a user directory. You can create a new user store or use an existing directory.

Create a separate user record for each affiliate site that retrieving assertions from the Identity Provider.

An attribute of the user record should have the same value that is specified in the Name field of the Service Provider Properties dialog box.

For example, if you identified the affiliate as Company A in the Name field, the user directory entry should be:

uid=CompanyA, ou=Development, o=partner

The Policy Server will map the subject DN value of the Service Provider's client certificate to this directory entry.

2. Add the configured user directory to the FederationWebServicesDomain.
3. Create a certificate mapping entry.

The value for the Attribute Name field in the Certificate Mapping Properties dialog box should be mapped to the user directory entry for the Service Provider. The attribute represents the subject DN entry in the Service Provider's certificate. For example, you may select CN as the Attribute Name, and this represents the Service Provider named cn=CompanyA,ou=Development,o=partner

4. Configure an X509 Client Certificate authentication scheme.

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

- Name: *<any\_name>*  
Example: cert artifact resolution
- Agent: FederationWebServicesAgentGroup
- Resource Filter: /affwebservices/saml2certartifactresolution
- Authentication Scheme: client cert auth scheme created in the previous step.

6. Create a rule under the cert artifact resolution realm containing the following:

- Name: *<any\_name>*  
Example: cert artifact resolution rule
- Resource: \*
- Web Agent Actions: GET, POST, PUT

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

The Artifact Resolution Service uses this HTTP header to make sure that the Service Provider for which the SAML assertion was generated is the one actually 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 Service Provider name value. For example, the entry could be uid=CompanyA.

Based on these entries, the Web Agent will return 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: *<responsecreated\_earlier\_in\_this\_procedure>*

9. Complete the configuration steps at the Service Provider to use client certificate authentication, if they are not completed already.



# Chapter 13: Configure SAML 2.0 Affiliations At the Identity Provider

---

This section contains the following topics:

[Affiliation Overview](#) (see page 349)

[Configure Affiliations](#) (see page 350)

## Affiliation Overview

A SAML affiliation is a group of SAML entities that share a name identifier for a single principal.

Both Service Providers and Identity Providers can belong to an affiliation; however, an entity may belong to no more than one affiliation. Service Providers share the Name ID definition across the affiliation. Identity Providers share the user disambiguation properties across the affiliation.

Using affiliations reduces the configuration 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 use cases:

- Single sign-on
- Single logout

Affiliations are set up at the Identity Provider site. Service Providers are added to an affiliation at the Service Provider site.

**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 verifies that the Service Provider is a member of the affiliation identified in the AuthnRequest, and it generates the assertion with the Name ID shared by the affiliation. It 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 checks if the Identity Provider belongs to an affiliation and sets an attribute in the request to the affiliation's ID. The Identity Provider receives the request and checks that the Service Provider belongs to the affiliation identified in the attribute.

The Identity Provider obtains the affiliation Name ID from the Session Server's 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

#### More Information:

[SAML Affiliation Dialog Fields and Controls](#) (see page 584)

### 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.

**More Information:**

[SAML Affiliation Dialog--Name IDs Tab](#) (see page 585)

## 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.

**More Information:**

[SAML Affiliation Dialog--Users Tab](#) (see page 586)

## 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 350)

## 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 353)



# Chapter 14: Configure SiteMinder as a SAML 2.0 Service Provider

---

This section contains the following topics:

[SAML 2.0 Authentication Scheme Overview](#) (see page 353)  
[Configuration Tasks for SAML 2.0 Authentication](#) (see page 356)  
[SAML 2.0 Authentication Scheme Prerequisites](#) (see page 357)  
[Configure the SAML 2.0 Authentication Scheme](#) (see page 359)  
[Configure User Disambiguation for User Look Ups](#) (see page 360)  
[Configure Single Sign-on at the SP](#) (see page 364)  
[Supply SAML Attributes as HTTP Headers](#) (see page 370)  
[Request Processing with a Proxy Server at the SP](#) (see page 376)  
[Enable Single Logout](#) (see page 377)  
[How To Protect Resources with a SAML 2.0 Authentication Scheme](#) (see page 378)  
[Enforce Assertion Encryption Requirements for Single Sign-on](#) (see page 384)  
[Customize Assertion Processing with the Message Consumer Plug-in](#) (see page 385)  
[Specify Redirect URLs for Failed SAML 2.0 Authentication](#) (see page 388)  
[Access the Artifact Resolution Service with a Client Certificate \(optional\)](#) (see page 390)

## SAML 2.0 Authentication Scheme Overview

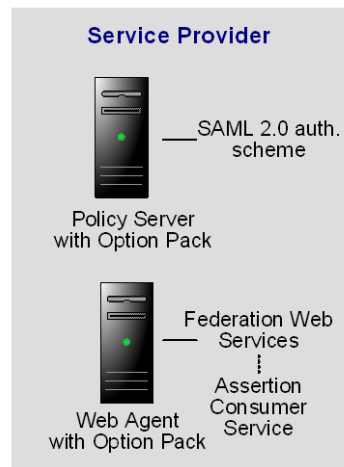
If you installed the Policy Server and Policy Server Option Pack or SiteMinder SPS federation gateway at a consumer site, any SiteMinder site can consume assertions to authenticate and authorize users. If you have sites in your federated network that have user stores, you may want to use SAML 2.0 authentication.

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 SiteMinder Federation Web Services application functions. For information about installing and configuring the SPS federation gateway, see the *CA SiteMinder Secure Proxy Server Administration Guide*.

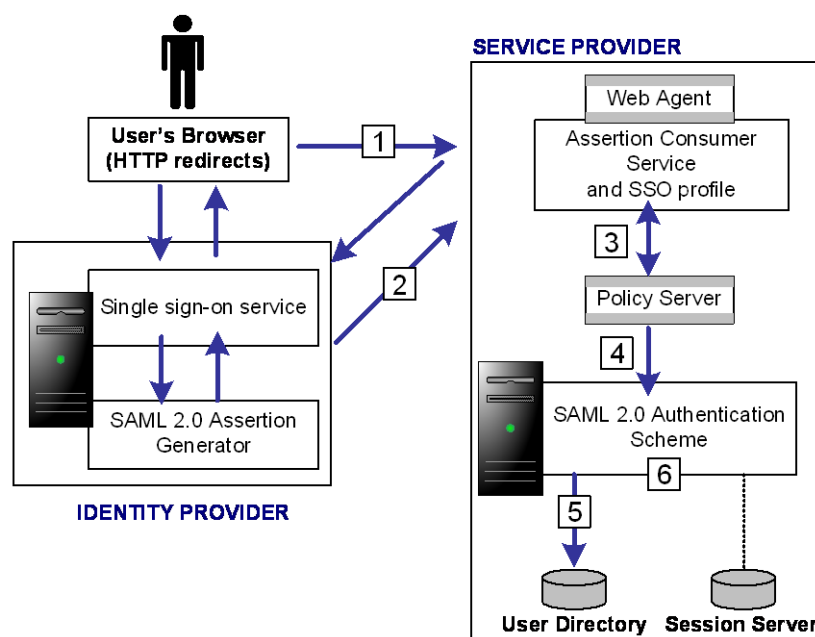
The SAML 2.0 authentication scheme is configured at the Service Provider's Policy Server, and is invoked by the Assertion Consumer Service. This service is a component of the Federation Web Services application and is installed on the Service Provider's Web Agent or SPS federation gateway. The Assertion 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 SiteMinder Federation Web Services application functions. For information about installing and configuring the SPS federation gateway, see the *CA SiteMinder Secure Proxy Server Administration Guide*.

The functional flow for authentication is as follows:

1. A user's browser 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.

In the case of the 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.

If the artifact binding is being used, the Assertion Consumer Service sends the artifact to the Identity Provider to obtain a response that contains the actual assertion. The Assertion Consumer Service sends 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, validates the assertion itself for time validity, and, if applicable, verifies if the assertion was signed by a trusted Identity Provider.

**Note:** For the POST binding, a signature is required and there must be certificate lookup information supplied. If a signature is not present, authentication fails. However, for the Artifact binding, a signed assertion is optional because the assertion response is obtained over a secure channel between the Service Provider and Identity Provider.

If Single Logout is enabled, the user is redirected by the SLO servlet to a No Access URL.

**More Information:**

[Configure User Disambiguation for User Look Ups](#) (see page 360)

## Configuration Tasks for SAML 2.0 Authentication

**Required Tasks**

| Check Here | Required Tasks                                                                                                                                         |
|------------|--------------------------------------------------------------------------------------------------------------------------------------------------------|
|            | Complete the SAML 2.0 authentication scheme prerequisites.                                                                                             |
|            | Set-up the authentication scheme for each Identity Provider that generated assertions. Assign a name, description, scheme type, and protection level.  |
|            | Specify the users who will be authenticated using the SAML 2.0 authentication scheme.                                                                  |
|            | Configure a single sign-on and select the binding to be used.                                                                                          |
|            | Associate that scheme with a realm. You can do this on a per Identity Provider basis or create a single custom authentication scheme and single realm. |

**Optional Tasks**

| Check Here | Optional Tasks        |
|------------|-----------------------|
|            | Enable single logout. |

| Check Here | Optional Tasks                                                         |
|------------|------------------------------------------------------------------------|
|            | 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.               |

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 503).
- To ensure 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 623).

## SAML 2.0 Authentication Scheme Prerequisites

There are several prerequisites you must fulfill before configuration a SAML authentication scheme.

### Install the Policy Server and Its Option Pack

The Policy Server together with the Policy Server Option Pack provides the SAML authentication scheme at the Service Provider. It also provides the SAML assertion generator used by a producer.

**Note:** Install these components at the Identity Provider and Service Provider.

For installation instructions, refer to the following guides:

- *Policy Server Installation Guide*
- *Policy Server, Web Agent Option Pack Guide*

### Install the Web Agent or SPS Federation Gateway

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 the SPS federation gateway, which has an embedded Web Agent.

**Note:** Install this component at the Identity Provider and Service Provider.

**At the consuming authority, set up the following components**

1. Install one of the following:
  - Web Agent
  - SPS federation gateway
2. Configure the Web Agent or SPS federation gateway.
  - For instructions about the Web Agent, see the *Web Agent Installation Guide* and *Web Agent Guide*.
  - For instructions about the SPS federation gateway, see the *Secure Proxy Server Administration Guide*.

**Important!** You must define a value for the Web Agent configuration parameter `DefaultAgentName` for all Service Provider Web Agents.

3. (Web Agent configuration only) If you are using a Web Agent at the Service Provider, you must specify a value for the Web Agent configuration parameter `DefaultAgentName` for all Service Provider Web Agents. The `DefaultAgentName` parameter specifies a Web Agent identity. Additionally, the specified Agent identity must be included in the Resource Filter of the realm that protects the target resource. You configure the `DefaultAgentName` parameter in the Agent Configuration Object or the local Agent configuration file.

**Note:** Omitting the `DefaultAgentName` parameter or using the value specified in the `AgentName` parameter in the realm resource filter causes SAML 2.0 authentication to fail, regardless of the single sign-on profile.

## Set Up a Key Database to Sign and Verify SAML POST Responses

SiteMinder can sign and verify SAML POST Responses and sign AuthnRequest Messages.

To use SAML POST profile for passing assertions, the assertion generator at the Identity Provider uses its private key and signs the SAML response that contains the assertion. The Service Provider then needs to verify that signature using public-key certificates.

In addition to the response being signed, you can sign an AuthnRequest message. The AuthnRequest message is sent during the authentication process to authenticate a user for cross-domain single sign-on.

To accomplish these tasks, you must set up a key database for each Policy Server that is responsible for signing, verification or both.

**More Information:**

[SmKeyDatabase Overview](#) (see page 479)

## 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](#) (see page 357).
2. Log in to the Policy Server User Interface.
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.
- 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.

**More Information:**

[SAML 2.0 Auth Scheme Properties Dialog Fields and Controls](#) (see page 568)

## 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`.

## Configure User Disambiguation for User Look Ups

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 value in the assertion; however, you can also obtain the LoginID by specifying 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, if 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 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.

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 362)  
[Use a SAML Affiliation to Locate a User Record \(Optional\)](#) (see page 361)

## 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 will use that same name ID, thereby 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 choose 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:** An affiliation has to be defined before you can select it.

**To select an affiliation**

1. From the Authentication Scheme Properties dialog box, click Additional Configuration.  
The SAML 2.0 Auth Scheme Properties dialog box opens.
2. Select the Users tab.
3. In the SAML Affiliation drop-down field, select a pre-defined 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 349)  
[SAML 2.0 Auth Scheme Properties Dialog--Users Tab](#) (see page 569)

## Configure Disambiguation Locally as Part of the Authentication Scheme

If you choose to disambiguate locally, there are two steps in the process:

1. Obtain the LoginID by the default behavior or by using an Xpath query.
2. Locate the user in the user store by the default behavior or 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 should not contain namespace prefixes. The following 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.

**More Information:**

[SAML 2.0 Auth Scheme Properties Dialog--Users Tab](#) (see page 569)

## Use a Search Specification to Locate a User

After obtaining 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 box, click Additional Configuration.

The SAML 2.0 Auth Scheme Properties dialog box 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 box 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 checked against the user store to find the correct record for authentication.

5. Click OK to save your configuration changes.

**More Information:**

[SAML 2.0 Auth Scheme Properties Dialog--Users Tab](#) (see page 569)

## Configure Single Sign-on at the SP

To establish single sign-on between the Identity Provider and the Service Provider, you need to 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, Federation Security Services 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, Federation Security Services 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 box, click Additional Configuration.

The SAML 2.0 Auth Scheme Properties dialog box 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 choose HTTP-Artifact as the binding, you must 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 group box, you can select both HTTP-Artifact and HTTP-Post.

If HTTP-POST is selected and artifact is not selected, only the POST binding will be accepted from the Identity Provider. If no binding is specified, the default is HTTP-artifact.

If you select HTTP-Artifact binding, you have to:

- 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 207).
- 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 365).
- 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 will be 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:**

[Protect the Assertion Retrieval or Artifact Resolution Service \(optional\)](#) (see page 236)  
[SAML 2.0 Auth Scheme Properties Dialog--SSO Tab](#) (see page 571)

## 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 Policy Server User Interface.
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:**

[SAML 2.0 Auth Scheme Properties Dialog--Backchannel Tab](#) (see page 578)

[WebLogic Configuration Required for Back Channel Authentication](#) (see page 366)

## WebLogic Configuration Required for Back Channel Authentication

If you installed the Web Agent Option Pack on a WebLogic 9.2.x application server at the Identity Provider, you have to set the <enforce-valid-basic-auth-credentials> element to false for Basic authentication across the artifact back channel to work.

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>
```

## Enforcing a Single Use Policy to Enhance Security

The single use policy feature prevents SAML 2.0 assertions that arrive via POST binding from being re-used at a Service Provider to establish a second session.

**Note:** Single use policy feature is enabled by default when you select the HTTP-POST binding.

Ensuring that an assertion is used only one time is an additional security measure for authenticating across a single sign-on environment. It mitigates security risks caused when an attacker acquires a SAML assertion from a user's browser that has already 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.

A single use policy is enabled by a storage mechanism provided by the SiteMinder Session Server. This mechanism is expiry data. Expiry data ensures a single use policy for SAML 2.0 POST-binding assertions by storing time-based data about an assertion. The SAML 2.0 authentication scheme uses the expiry data interface to access the expiry data in the Session Server database.

## How the Single Use Policy is Enforced

Upon successful validation of a SAML 2.0 assertion, the SAML 2.0 authentication scheme writes assertion data in the expiry data table with a key of the assertion ID and an expiration time. The Session Server Management thread in the Policy Server deletes expired data from the expiry data table.

If single policy use is enforced, writing assertion data will fail if an entry already exists in the expiry data table with a key of the assertion ID because the assertion has already been used to establish a session. If the scheme cannot write to the table in the session server, the SAML 2.0 authentication scheme denies the authentication in the same manner as an invalid assertion.

Writing assertion data may fail for other reasons; however, if the single use of the assertion cannot be enforced because the database is unavailable for any reason, then the authentication scheme will deny the request to ensure that assertions cannot be re-used.

## 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 366)

[Storing User Session, Assertion, and Expiry Data](#) (see page 207)

## Permit the Creation of a Name Identifier for SSO

As part of a single sign-on request, a Service Provider may request a particular user attribute to be included in the assertion at the Identity Provider; however, the value of the required attribute may not be available in the user record at the Identity Provider.

You can enable the Service Provider to include the Allow/Create attribute as part of its AuthnRequest message to the Identity Provider. When a request is received at the Identity Provider, this attribute, together with the corresponding feature enabled at the Identity Provider, instructs the Identity Provider to generate a new, unique value for the NameID if it cannot find the requested attribute in its user store. This value is then included in the assertion 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, if the Service Provider locates the user record in its user store, it grants the user access.

If the Identity Provider is not configured to create a new identifier, it will not generate a unique identifier, regardless of whether an Allow/Create attribute is part of an AuthnRequest message. In this case, the normal flow of assertion generation continues after an entry is made in the Identity Provider log files that a unique identifier was not created.

For a unique identifier to be generated, the Allow/Create feature must be configured at both sites.

### More Information:

[SAML 2.0 Auth Scheme Properties Dialog--SSO Tab](#) (see page 571)

## 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.



## Enable the Enhanced Client or Proxy Profile

The Enhanced Client or Proxy Profile (ECP) is an application of the single sign-on profile. An ECP is a system entity that knows how to contact an Identity Provider and supports the Reverse SOAP binding, PAOS for the purpose of providing single sign-on for a user.

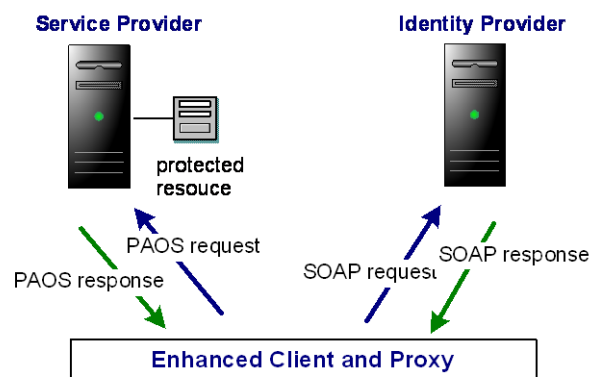
An enhanced client may be a browser or some other user agent that supports the ECP functionality. An enhanced proxy is an HTTP proxy, such as a Wireless Access Protocol proxy for a wireless device.

The ECP profile allows the Service Provider to make an authentication request without knowing the Identity Provider, and PAOS lets the Service Provider obtain the assertion through the ECP, which is always directly accessible, unlike the Identity Provider. The ECP acts as the intermediary between the Service Provider and the Identity Provider.

You might want to enable the ECP profile with single sign-on in the following situations:

- For a Service Provider that expects to service enhanced clients or proxies that require this profile.
- When the Identity Provider and Service Provider cannot communicate directly.
- When a proxy server is in use, such as a wireless access protocol (WAP) gateway in front of a mobile device with limited functionality

The flow of the ECP profile is shown in the following illustration.



**To enable the ECP profile**

1. Make sure the ECP request is directed to the AuthnRequest service. The following URL shows an example:

`https://host:port/affwebservices/public/saml2authnrequest=`

2. Make sure that the headers in the ECP request include attributes required by the [SAML 2.0 specification](#), such as the following:

```
Accept: text/html; application/vnd.paos+xml
PAOS: ver='urn:liberty:paos:2003-08' ;
      'urn:oasis:names:tc:SAML:2.0:profiles:SSO:ecp'
```

3. Use the Policy Server User Interface to fill out the required single sign-on fields on the SSO tab.
4. Select the Enhanced Client and Proxy Profile check box.
5. Click OK.

**More Information:**

[SAML 2.0 Auth Scheme Properties Dialog--SSO Tab](#) (see page 571)

## Supply SAML Attributes as HTTP Headers

An assertion response may include attributes in the assertion. These attributes can be supplied as HTTP header variables and used by a client application can use these headers for finer grained access control.

The benefits of including attributes in HTTP headers is as follows:

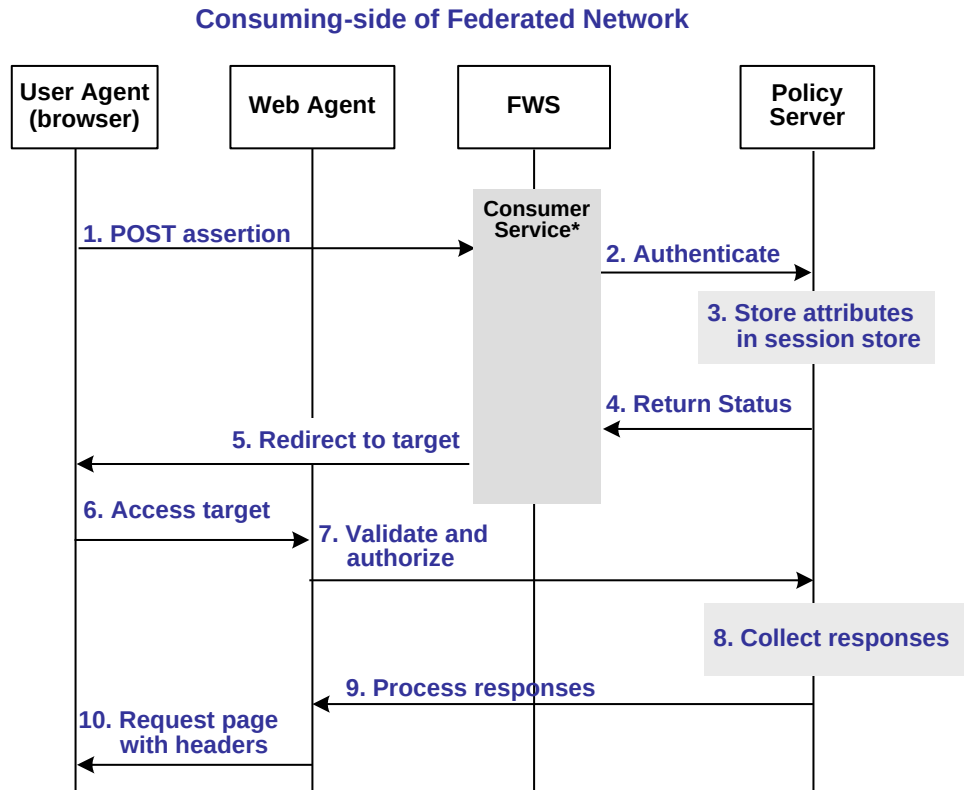
- HTTP headers are not persistent. They are present only within the request or response that contains them.
- HTTP headers, as supplied by the SiteMinder Web Agent, are not seen by the user's browser, which reduces security concerns.

**Note:** The HTTP headers have size restrictions that the attributes cannot exceed. Federation Security Services 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's application.

The following flow diagram shows the sequence of events at runtime:



\*Consumer service can be one of the following:

- SAML Credential Collector (SAML 1.x)
- Assertion Consumer Service (SAML 2.0)
- Security Token Consumer Service (WS-Federation)

**Note:** The SPS federation gateway can replace the Web Agent and Web Agent Option Pack to provide the SiteMinder 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 *CA SiteMinder 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 producing partner, it sends the assertion to the appropriate consumer service at the consuming partner. 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's 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's session and to ensure 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

There are several configuration steps required to retrieve the SAML attributes cached in the session store and provide them as HTTP headers.

**The components you must configure are as follows**

1. Select PersistAttributes as the redirect mode for the SAML authentication scheme. This enables the SAML Attributes to be returned as HTTP headers.
2. Configure an authorization rule for the realm that contains the target resource.
3. Configure a response that uses the active response type for each SAML attribute to be supplied as a header.
4. 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 Policy Server User Interface.
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, you need to 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, therefore the Web Agent cannot re-authenticate the user and pass on the HTTP headers. So, the retrieval of the attributes must happen during the authorization stage.

### To create an OnAccessAccept Rule for the realm

1. Log on to the Policy Server User Interface.
2. From the Domains tab, navigate to the realm which protects the target resource.
3. Select the realm with the target resource and choose 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. Choose 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 group box..

8. Ensure that Enabled is checked in the Allow/Deny and Enable/Disable group box.
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

You must configure a response that sends the SAML attributes as HTTP headers to the Web Agent. The Web Agent will process the response and make the header variables available to the client application.

### To create a response to send the attributes as headers

1. Log on to the Policy Server User Interface.
2. From the Domains tab, select the domain that contains the target resource.
3. In the domain, select the Responses object and create a new 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 group box.
7. Complete the fields in the Attribute Fields group box as follows:

#### Variable Name

Specify the name you want for the header variable. You assign this name.

#### Library Name

smfedattrresponse

This must be the entry for this field.

#### Function Name

getAttributeValue

This 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 will be in the assertion.

8. Click on OK to save the attribute.
9. Repeat the procedure for each attribute that should become an HTTP header variable. You can configure many attributes for a single response.

The response will send 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 Policy Server User Interface.
2. From the Domains tab, select the domain that contains the target resource.
3. In the domain, select the Policies object and create a new policy.  
The Policy Properties dialog opens.
4. Enter a descriptive name in the Name field.
5. Select the users that should 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.

## Request Processing with a Proxy Server at the SP

When SiteMinder receives certain requests at the SP, it validates the message attributes using the local URL for Federation Web Services application before processing the request.

For example, a logout request message may 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.

When the SiteMinder federated environment sits behind a proxy server, the local and destination attribute URLs are not the same because the Destination attribute is the URL of the proxy server. For example, the logout message may 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. 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.

## 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 Policy Server User Interface.
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 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_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 Single Logout

The single logout (SLO) profile allows near-simultaneous logout of all sessions provided by a specific session authority and associated with a particular user. The logout is directly initiated by the user. A session authority is the authenticating entity that has initially authenticated the user. In most cases, the session authority is the Identity Provider.

The benefit to configuring single logout is that it ensures no sessions are left open for unauthorized users to gain access to resources at the Service Provider.

The single logout service can be initiated via user's browser from a link at the Service Provider or at the Identity Provider. When a user wants to logout, he clicks the logout link which points to an SLO servlet. This servlet, which is a component of Federation Web Services, processes logout requests and response coming from a Service Provider or Identity Provider; however, the servlet does not need to know the originator of the request or response. It uses the SiteMinder session cookie to determine the session to log out.

## Bindings for Single Logout

The single logout feature transports messages using the HTTP-Redirect binding. This binding determines how SAML protocol messages are transported using HTTP redirect messages, which are 302 status code responses.

## Configure Single Logout

### 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:

[SAML 2.0 Auth Scheme Properties Dialog--SLO Tab](#) (see page 575)  
[Storing User Session, Assertion, and Expiry Data](#) (see page 207)

## How To Protect Resources with a SAML 2.0 Authentication Scheme

At the Service Provider, you must configure a SAML authentication scheme for each Identity Provider that generates assertions. Each scheme must be bound to a realm, which consists of all the target URLs that comprise the target resources requested by users. These resources then need to be protected by a SiteMinder policy.

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 being requested by users.

There are two ways to set-up a realm that includes a SAML authentication scheme:

- You can create a unique realm for each authentication scheme and Identity Provider already configured.
- You can configure a single target realm 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 configuring a realm, 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 Identity Provider to the Service Provider, without an initial request from the Service Provider. In this response is the target. At the Identity Provider site, an administrator needs to include this response in a link so that this link the user gets redirected to the Service Provider.

## Assign a SAML 2.0 Authentication Scheme to a Realm

### Configure a Unique Realm for Each SAML Authentication Scheme

The process for configuring a unique realm for each SAML authentication (artifact or profile), follows the standard instructions for creating realms using the Policy Server User Interface.

### Configure a Single Target Realm for All SAML Authentication Schemes

To simplify configuration of realms for all SAML authentication schemes, you create a single target realm for multiple Identity Providers. To do this, set-up:

- A single custom authentication scheme  
You should have already configured a SAML 2.0 authentication scheme for each Identity Provider prior to configuring a custom template.
- A single realm with one target URL

### Configure a Custom Authentication Scheme for the Single Realm

#### To create a custom authentication scheme for a single realm

1. Log into the Policy Server User Interface.
2. From the menu bar, select Edit, System Configuration, Create Authentication Scheme.  
The Authentication Scheme Properties dialog box opens.
3. From the Authentication Scheme Type drop-down list, select Custom Template. The contents of the dialog box change for the custom template.
4. In the Library field, enter `smauthsinglefed` for the library name. This is the library name for the custom single authentication scheme.

5. Disregard the Secret and Confirm Secret fields.

In the Parameter field, which instructs the custom scheme which SAML authentication schemes it should use, specify one of the following options:

- SCHEMESET=LIST; <saml-scheme1>;<saml\_scheme2>

Specifies the list of SAML authentication scheme names to use. If you configured an artifact scheme called artifact\_Idp1 and POST profile scheme called samlpst\_IdP2, you will enter these schemes.

- 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.

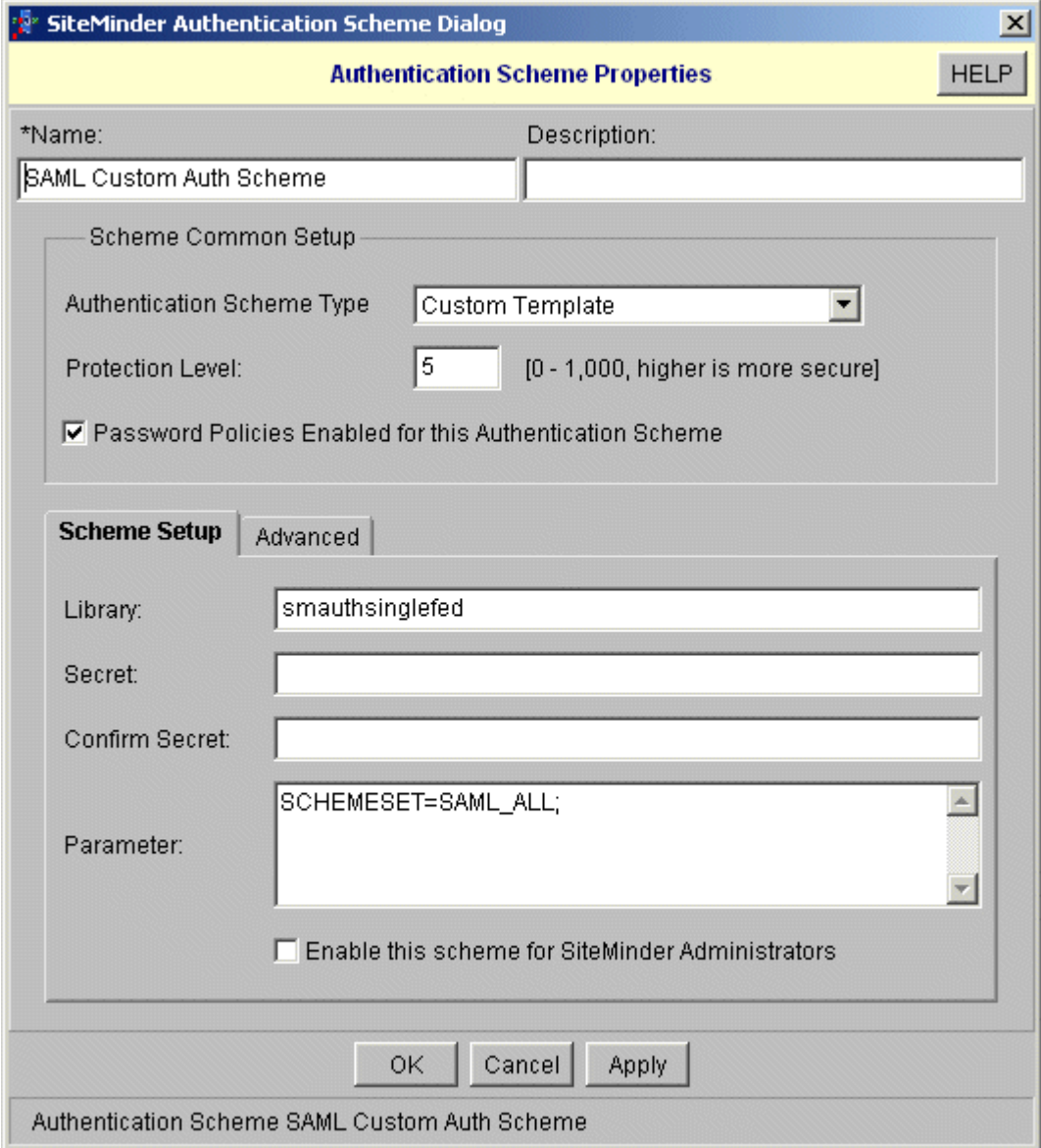
- SCHEMESET=SAML\_POST;

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.

6. Disregard the Enable this scheme for SiteMinder Administrators checkbox.
7. Click OK to save your changes.



The image shows the 'SiteMinder Authentication Scheme Dialog' with the 'Authentication Scheme Properties' tab selected. The dialog has a title bar with a close button. Below the title bar is a yellow header with the text 'Authentication Scheme Properties' and a 'HELP' button. The main area is divided into two sections: 'Scheme Common Setup' and 'Scheme Setup'. The 'Scheme Common Setup' section contains fields for '\*Name:' (SAML Custom Auth Scheme) and 'Description:', a dropdown for 'Authentication Scheme Type' (Custom Template), a 'Protection Level' (5) with a note '[0 - 1,000, higher is more secure]', and a checked checkbox for 'Password Policies Enabled for this Authentication Scheme'. The 'Scheme Setup' section has two tabs: 'Scheme Setup' (selected) and 'Advanced'. It contains fields for 'Library:' (smauthsinglefed), 'Secret:', 'Confirm Secret:', and 'Parameter:' (SCHEMESET=SAML\_ALL;). There is also an unchecked checkbox for 'Enable this scheme for SiteMinder Administrators'. At the bottom are 'OK', 'Cancel', and 'Apply' buttons. A status bar at the very bottom reads 'Authentication Scheme SAML Custom Auth Scheme'.

**SiteMinder Authentication Scheme Dialog**

**Authentication Scheme Properties** HELP

\*Name: SAML Custom Auth Scheme Description:

**Scheme Common Setup**

Authentication Scheme Type: Custom Template

Protection Level: 5 [0 - 1,000, higher is more secure]

☒ Password Policies Enabled for this Authentication Scheme

**Scheme Setup** Advanced

Library: smaauthsinglefed

Secret:

Confirm Secret:

Parameter: SCHEMESET=SAML\_ALL;

☐ Enable this scheme for SiteMinder Administrators

OK Cancel Apply

Authentication Scheme SAML Custom Auth Scheme

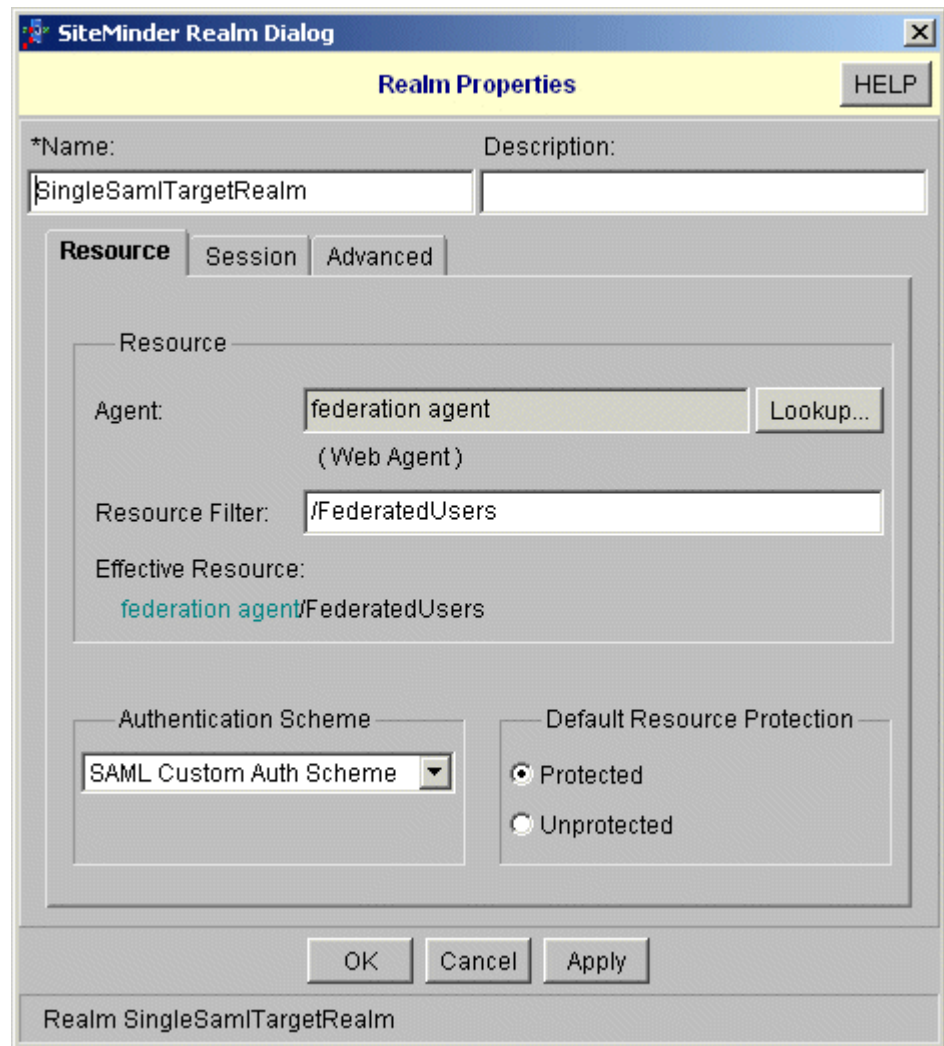
## Configure a Single Target Realm

### To create the single target realm

1. Log into the Policy Server User Interface.
2. Display the policy domains.
3. Expand the policy domain where you will add the new realm.

4. Click on the realm icon.
5. From the menu bar, select Edit, Create Realm.  
The SiteMinder Realm dialog box opens.
6. In the Name field, enter a name for this custom target realm.
7. In the Agent field, select a SiteMinder Web Agent protecting the Web server with the target Service Provider resource.
8. In the Resource Filter field, specify the location of the target resource.  
For example, /FederatedUsers
9. From the Authentication Scheme drop-down list, select the custom authentication scheme that you configured for directing requests to the appropriate SAML authentication schemes.
10. Ensure that Protected is selected in the Default Resource Protected group box.
11. Click OK to save the realm configuration.

The following graphic shows the Realm Properties dialog box for the custom realm.



## Create a Rule for the Target Resource

After adding the target resources to a realm, create rules for the realms that protect either the entire contents of the realm or a specific resource in the realm.

## Create a Policy for the Target Resource

### To protect the target resource

1. From the System tab, create a policy domain.  
Do not create an affiliate domain.
2. Under the policy domain, create a realm that contains the target resources and select the SAML 2.0 authentication scheme (artifact or post) for this realm.
3. Under the realm, create a rule for the resources in the realm and select the Web Agent actions.
4. Create a policy. This policy should contain:
  - users that should have access to the target resources.
  - rule associated with the realm that contains the target resources

## Enforce Assertion Encryption Requirements for Single Sign-on

The encryption feature ensures that the authentication scheme processes only an encrypted assertion and/or Name ID in the assertion.

For added security, the Identity Provider may have encrypted the Name ID, user attributes, and/or the entire assertion. Encryption adds another level of protection when transmitting the assertion. When encryption is enabled at the Identity Provider, the 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 the encryption at the Session Provider, the assertion must contain an encrypted Name ID and/or assertion or the Service Provider will not accept 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.

**More Information:**

[SAML 2.0 Auth Scheme Properties Dialog--Encryption Tab](#) (see page 577)

## Customize Assertion Processing with the Message Consumer Plug-in

The Message Consumer Plug-in is SiteMinder's Java program that implements the Message Consumer Extension API. Using this plug-in you can implement your own business logic for processing assertions, such as rejecting an assertion and returning a SiteMinder-defined status code. This additional processing works together with SiteMinder's 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 is configured. The plug-in then has the opportunity to disambiguate the user, if it knows how.

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 should return a `UserNotFound` error, which is documented in the `MessageConsumerPlugin` interface. The plug-in's use of SiteMinder's redirect URLs feature is optional and is based on the error code returned by the plug-in. If the Message Consumer plug-in is not configured, the redirect URLs are used based on the error generated by the SAML authentication scheme.

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, but this is not required.

To integrate the Message Consumer plug-in with SiteMinder, use the parameter values that you specify for the plug-in configuration. The plug-in configuration is part of the SAML 1.x, SAML 2.0 and WS-Federation authentication scheme configuration.

**More information:**

[Specify Redirect URLs for Failed SAML 1.x Authentication](#) (see page 293)

[Specify Redirect URLs for Failed SAML 2.0 Authentication](#) (see page 388)

[Specify Redirect URLs for Failed WS-Federation Authentication](#) (see page 449)

## Configuring the Message Consumer Plug-in (SAML 2.0)

**To configure the message consumer plug-in**

1. From the Authentication Scheme Properties dialog box, click Additional Configuration.

The SAML 2.0 Auth Scheme Properties dialog box opens.

2. Select the Advanced tab.
3. Implement the plug-in class.

A sample class, `MessageConsumerSAML20.java`, can be found in `sdk/samples/authextensionsaml20`.

4. In the Full Java Class Name field, enter the Java class name of the plug-in. This plug-in is invoked by the Message Consumer at run time.

The plug-in class can parse and modify the assertion, and then return the result to the Message Consumer for final processing.

Only one plug-in is allowed for each authentication scheme. For example, `com.mycompany.messageconsumer.SampleCode`

A sample plug-in is included in the SDK. You can view a sample message consumer plug-in at `sdk/samples/authextensionsaml20`.

**Note:** Specify a Message Consumer plug-in for each authentication scheme.

## Integrate the Message Consumer Plug-in with SiteMinder (SAML 2.0)

After configuring a message consumer plug-in, you have to integrate the plug-in with the SAML 2.0 authentication scheme.

The instructions for compiling the message consumer plug-in Java file are in the AssertionSample.java file, in sdk/samples/authextensionsaml20.

### To integrate the Message Consumer plug-in with the authentication scheme

1. Compile the message consumer plug-in Java file.

The Java file requires the following .jar file installed with the Policy Server:

`policy_server_home/bin/java/SmJavaApi.jar`

2. In the JVMOptions.txt file, modify the -Djava.class.path value so it is set to the classpath for the plug-in. This enables the plug-in to be loaded.

**Note:** Do not modify the classpath for SMJavaApi.jar.

3. Restart the Policy Server.

Restarting ensures that the latest version of the message consumer plug-in is picked up after being recompiled.

**Note:** Instead of specifying the message consumer plug-in class and its parameters via the Policy Server User Interface you can use the Policy Management API (C or Perl). For instructions, see the *SiteMinder Programming Guide for C* or the *SiteMinder Programming Guide for Java*.

Additional information about the Message Consumer 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 *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*.

## 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.

## Access the Artifact Resolution Service with a Client Certificate (optional)

This procedure is only for single sign-on with the artifact binding.

You can use client certificate authentication to secure the back-channel across which the Identity Provider sends the assertion to the Service Provider when using the HTTP-artifact binding.

**Note:** Certificate authentication for the back-channel is optional; you can use Basic authentication instead.

The SAML 2.0 authentication scheme with artifact binding is invoked by the Assertion Consumer Service. This service collects information from the scheme to retrieve the SAML assertion from the Identity Provider. You are required to specify an authentication method for the realm that contains the Artifact Resolution Service at the Identity Provider. This tells the Assertion Consumer Service what type of credentials to provide to retrieve the assertion.

If the Artifact Resolution Service is part of a realm configured for client certificate authentication, there are some configuration tasks at the Service Provider and the Identity Provider you need to complete.

- At the Service Provider, you need to select the client certificate option as part of the authentication scheme configuration as instructed in [Configuring the Client Certificate Option at the Service Provider](#).
- At the Identity Provider, you need to create a policy to protect the Artifact Resolution Service as instructed in [Protecting the Artifact Resolution Service at the Identity Provider](#).

### Configuring the Client Certificate Option at the Service Provider

To set-up the client certificate authentication to secure the backchannel to the artifact resolution service, you need to

- Select the client cert option in the authentication scheme configuration
- Add a client certificate to the smkeydatabase

### Select the Client Cert Option for Authentication

**To present a client certificate as credentials**

1. In the Authentication Scheme Properties dialog for SAML 2.0 authentication, click Additional Configuration.
2. Select the SSO tab.

3. Select HTTP-Artifact in the Bindings group box.
4. Select Client cert for the Authentication field.

## Add a Client Certificate to the SMKeyDatabase

This procedure assumes you already have a private key/certificate pair from a Certificate Authority.

1. Create an smkeydatabase, if one does not already exist. Enter the command:  
`smkeytool -createDB smkeydatabase -password <password>`
2. Add a private key and client certificate to smkeydatabase by entering the following command.

```
smkeytool -alias <alias> -addPrivKey - keyfile<file_path_to_key_file>  
-certfile<file_path_to_certificate>
```

Notes:

- The value for alias should be same as the value of the Name field specified in the Scheme Setup dialog for the SAML 2.0 authentication scheme with HTTP-artifact binding. The attribute of the Service Provider's subject DN, represented in the example by the CN value, should also reflect the Name value.

For example, if you entered CompanyA as the Name, then alias would be Company A, and the attribute could be CN=CompanyA, OU=Development, O=CA, L=Islandia, ST=NY, C=US

- To refer to the existing entry, subsequent keytool commands must use the same alias.
  - The value for keypass should be same as the value of the Password field specified in the Scheme Setup dialog for the SAML 2.0 authentication scheme.
3. Restart the Policy Server to see the changes to the smkeydatabase immediately.

## Protect the Artifact Resolution Service at the Identity Provider

At the Identity Provider Policy Server, you must [configure a policy to protect the artifact resolution service](#) (see page 345). The realm for this policy must use an X.509 client certificate authentication scheme.





# Chapter 15: Use an Attribute Authority to Authorize Users

---

This section contains the following topics:

[Perform Authorizations with an Attribute Authority](#) (see page 393)

[Flow Diagram for Authorizing a User with User Attributes](#) (see page 396)

[Configure an Attribute Authority and a SAML Requester](#) (see page 397)

[Set up the Attribute Authority](#) (see page 397)

[Set up a SAML Requestor to Generate Attribute Queries](#) (see page 400)

## Perform Authorizations with an Attribute Authority

The Policy Server can authorize a user based on the following types of information:

- The users to which the policy applies
- IP address restrictions
- Time restrictions
- Policy Expressions
- Active Policy

Additionally, the Policy Server can authorize a user based on user attributes provided by a SAML 2.0 Attribute Authority. When a user requests access to a protected resource, the authorizing entity can request additional user attributes to determine whether access to the resource should be granted.

In a SAML 2.0 federated network, there are two roles required to authorize a user based on user attributes:

- SAML Attribute Authority
- SAML Requester

### **SAML Attribute Authority**

The SAML Attribute Authority adheres to the SAML 2.0 Assertion Query/Request profile. It relies on the Attribute Service to process a query message and create attribute assertions. 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 retrieved and 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 an response message that includes an attribute assertion wrapped in a SOAP message. This response includes the user attributes.

**Note:** The user does not need to be authenticated at the Attribute Authority and there does not need to be a single sign-on relationship between the Authority and the Requester.

### SAML Requester

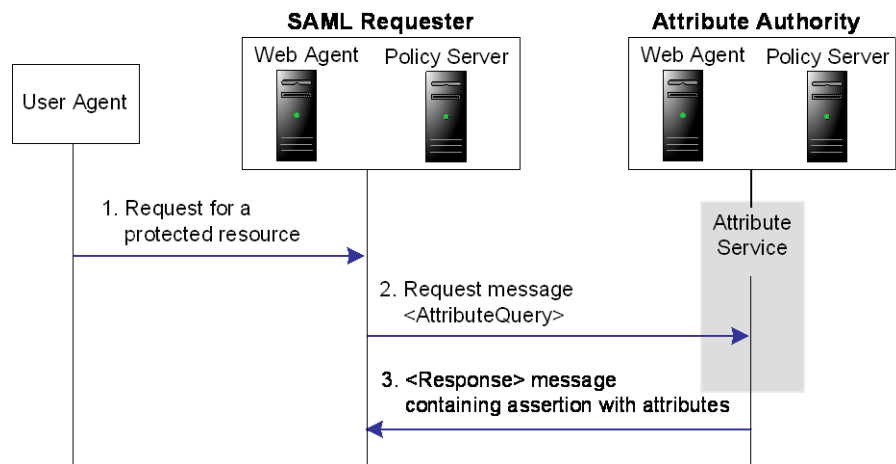
The SAML Requester is a SAML entity that uses the SAML 2.0 Assertion Query/Request profile to request attributes for a user. In SiteMinder, the SAML Requester is not a specific service, but a group of Policy Server features that can produce and process <AttributeQuery> messages. It is the Requester that 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 used by a policy expression.

**Note:** In a SiteMinder federated environment, the SAML Attribute Authority is the Identity Provider and the SAML Requester is the Service Provider; however, this does not have to be the case.

To evaluate an authorization request based on SAML 2.0 user attributes, you add a SiteMinder attribute type called a **federation attribute variable** to a policy expression that is used in a policy protecting the target resource at the SAML Requester. When this variable is used in a policy, 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 is required to have a session at the SAML Requester; however, the user does not have to log in or authenticate at the Attribute Authority.

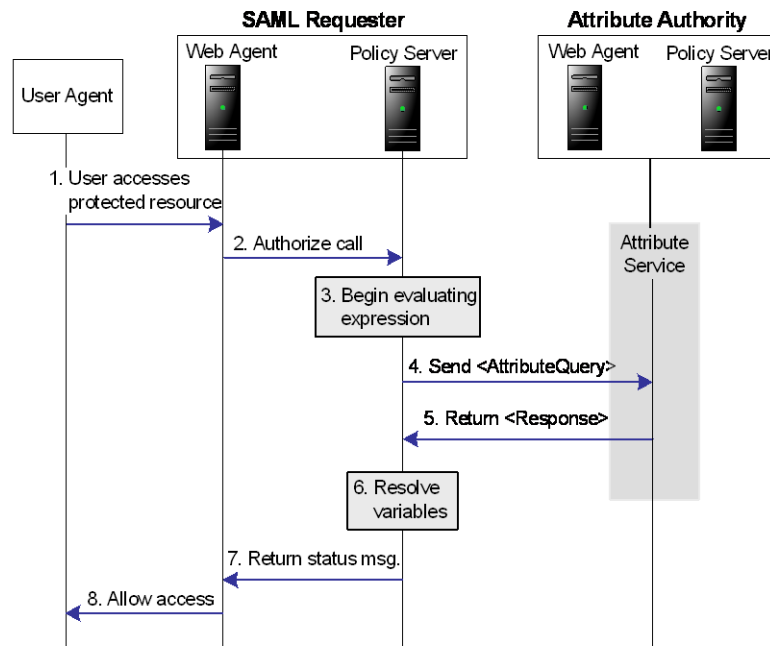
The following figure shows how an attribute query is processed.



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

The sequence of a user attribute request is as follows:

1. A user accesses a protected resource. The user might be logged in locally or may have authenticated via a SAML assertion.
2. The Web Agent at the SAML Requester makes a call to the local Policy Server determine if 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 the user up in the local user store to obtain the user's NameID.
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.

## Configure an Attribute Authority and a SAML Requester

### To configure the SAML Attribute Authority at the Identity Provider

1. Define a search specification for locating a user. Enter the NameID into the search specification.
2. Define the attributes that can be returned in response to a query message.

To configure the SAML Requester at the Service Provider to send an attribute query:

1. Enable the AttributeQuery functionality.
2. Configure the back channel across which communications between the SAML Requester.
3. Define the list of attributes that can be requested by the SAML Requester.
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 Policy Server User Interface.
2. From the appropriate affiliate domain, double-click the Service Provider, acting as the SAML Requester, that will be requesting user attributes.

The SAML Service Provider Properties dialog opens.

3. Select the Attribute Svc tab.
4. Check 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**

Check 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 box 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 checked 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 398).

**More Information**

[Attribute Authority and Query Reference](#) (see page 589)

## 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 attributes function is determined by the Retrieval Method field in the SAML Service Provider Attribute dialog.

If you want the same attribute to be used for both services, you must 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.

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 this attribute is requested by an attribute query, selecting Attribute Service as the Retrieval Method marks the attribute for inclusion in the attribute assertion.

#### More Information:

[Attribute Svc Dialog Fields and Controls](#) (see page 590)

## Configure the Back Channel for the Attribute Authority

If you configure a SAML Attribute Authority, you need to configure a secure backchannel across which the SAML Attribute Authority returns the SAML response to the requester.

**Note:** This procedure assumes you have already enabled the Attribute Authority Service.

#### To configure the backchannel

1. Open the Policy Server User Interface.
2. From the appropriate affiliate domain, double-click the Service Provider that will be requesting user attributes for authorization.

The SAML Service Provider Properties dialog opens.

3. Select the General tab.
4. Click on 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 may already have 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.

**More Information:**

[SAML Service Provider Dialog--Backchannel Properties](#) (see page 549)

## Set up a SAML Requestor to Generate Attribute Queries

For a Service Provider to act as a SAML Requester, you need to 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 Policy Server User Interface.
2. Display the Authentication Schemes object and double-click an existing SAML 2.0 authentication scheme or create a new scheme.

The Authentication Scheme Properties dialog opens.

3. Click Additional Configuration.

The SAML 2.0 Auth Scheme Properties dialog opens.

4. From this dialog you will configure fields in the following dialogs:

- Attributes tab
- NameIDs tab
- Backchannel tab

5. Finally, you will configure a Federation Attribute Variable.

**More Information:**

[Enable Attribute Queries and Specify Attributes](#) (see page 401)

[Configure the NameID for the Attribute Query](#) (see page 402)

[Configure the Backchannel for the Attribute Query](#) (see page 402)

[Create a Federation Attribute Variable](#) (see page 403)



## Enable Attribute Queries and Specify Attributes

### To enable the SAML Requester to send an attribute query

1. Log on to the Policy Server User Interface.
2. Access the Authentication Scheme Properties dialog for the SAML 2.0 authentication scheme that protects the resource that will be 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 group box, 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 402). This NameID configured in the SAML 2.0 Auth.Scheme Properties is included in the attribute query for use by the Attribute Authority.

### More Information:

[Attributes Dialog for Attribute Queries](#) (see page 592)

## 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 NameID is obtained by the SAML Requester 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.

You may be prompted to configure the backchannel if it is not already configured.

### More Information:

[Name IDs Dialog Fields for Attribute Queries](#) (see page 594)

[SAML 2.0 Auth Scheme Properties Dialog--Backchannel Tab](#) (see page 578)

## Configure the Backchannel for the Attribute Query

The attribute query is sent across a secure backchannel to the Attribute Authority.

**Note:** There is only one backchannel between the Service Provider and the Identity Provider. Therefore, the backchannel configuration that you define for the attribute query is the same backchannel configuration that is used if 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 on Additional Configuration

3. Select the Backchannel tab.
4. Complete the following fields:
  - Authentication
  - SP Name
  - Password
  - Confirm Password

**More Information:**

[SAML 2.0 Auth Scheme Properties Dialog--Backchannel Tab](#) (see page 578)

## Create a Federation Attribute Variable

To use a federation attribute variable in a policy expression, you need to first create the attribute variable.

**To define a federation attribute variable**

1. Log on to the Policy Server User Interface.
2. From the list of Domains, expand the policy domain where the variable will be added.
3. Expand the Variables list by clicking on 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.

**More Information:**

[Federation Attribute Variable Editor Reference](#) (see page 596)

## Create a Policy Expression with the Federation Attribute Variable

To use Federation attribute variables as part of the authorization process, you need to 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 16: Configure SiteMinder as an Account Partner

---

This section contains the following topics:

- [Configuration Checklist](#) (see page 405)
- [Add a Resource Partner to an Affiliate Domain](#) (see page 407)
- [Select Users for Which Assertions Will Be Generated](#) (see page 407)
- [Specify Name IDs for WS-Federation Assertions](#) (see page 410)
- [Configure Required General Information for WS-Federation](#) (see page 411)
- [Configure Single Sign-on for WS-Federation](#) (see page 412)
- [Configure Attributes for WS-Federation Assertions \(optional\)](#) (see page 417)
- [Configure Signout](#) (see page 422)
- [Validate Signout Requests that are Digitally Signed](#) (see page 423)
- [Customizing Content in WS-Federation Assertions](#) (see page 424)
- [Protect the Authentication URL to Generate a SiteMinder Session](#) (see page 427)

## Configuration Checklist

Identifying a Resource Partner in a SiteMinder federated network is a task you complete at the Account Partner because the Account Partner needs information about the Resource Partner to generate an assertion for that partner. At the Account Partner, you define how the two entities communicate determine the types of profiles that each side supports, such as Web single sign-on and signout.

- [Required Configuration Tasks](#) (see page 406)
- [Optional Configuration Tasks](#) (see page 406)

Tips:

- Certain parameter values at the Account Partner and Resource Partner must match for the configuration to work. A list of those parameters can be found in [Configuration Settings that Must Use the Same Values](#) (see page 503).
- 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 623).

## Required Configuration Tasks for Configuring Resource Partners

The following are required tasks for configuring Resource Partners:

- Create an Affiliate domain.
- Add a Resource Partner to the affiliate domain and:
  - Specify the name of the Resource Partner.
  - Specify and protect the Authentication URL.
  - Select users from a user store for which assertions will be generated.
  - Specify the Name ID on the Name IDs tab.
  - Specify the Resource Partner ID, the Account Partner ID, and the Skew Time on the General tab.
  - Configure all fields on the SSO tab to setup single sign-on.

You can save a Resource Partner entity without configuring a complete SSO profile; however, you cannot pass an assertion to the Resource Partner without configuring SSO.

### More Information:

[Creating Affiliate Domains](#) (see page 239)

[Add a Resource Partner to an Affiliate Domain](#) (see page 407)

## Optional Configuration Tasks for Configuring a Resource 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 signout.
- Customize a SAML response using the Assertion Generator plug-in.

### More Information:

[Specify IP Address Restrictions for Resource Partners \(optional\)](#) (see page 415)

[Set up Time Restrictions for Resource Partner Availability \(optional\)](#) (see page 416)

[Configure Signout](#) (see page 422)

[Customizing Content in WS-Federation Assertions](#) (see page 424)

## 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 Policy Server User Interface.
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.

### More Information:

[Protect the Authentication URL to Generate a SiteMinder Session](#) (see page 427)  
[WS-Federation Resource Partner Properties Reference](#) (see page 597)

## 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 may 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 into the Policy Server User Interface.
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 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.  

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 Policy Server User Interface 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.

**More Information:**

[Resource Partner Properties Dialog Fields and Controls](#) (see page 598)

## Excluding a User or Group from Resource Partner 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 Resource Partner, but you there is a small subset of this group that you want to exclude.

**To exclude a user or group from access to a Resource Partner'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 Resource Partner.
3. Click OK.



## Allow Nested LDAP Groups Resource Partner 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 Resource Partner 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 Resource Partner.

### To allow the Resource Partner 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 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 box, 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 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 Resource Partner.

2. Click Add to Current Members.

The Policy Server User Interface adds the user or query to the Current Members list.

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 Policy Server User Interface 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 Policy Server User Interface 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.

**More Information:**

[Resource Partner Properties Dialog--Name IDs Tab](#) (see page 602)

## 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 Policy Server User Interface.
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 252) 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:

[Resource Partner Properties Dialog--General Tab](#) (see page 603)

## 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 Policy Server User Interface.
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:**

[Resource Partner Properties Dialog--SSO Tab](#) (see page 604)

[Customizing Content in WS-Federation Assertions](#) (see page 424)

## Assertion Validity for Single Sign-on

For single sign-on, the values of the Skew Time and the Validity Duration determine how SiteMinder FSS calculates the total time that an assertion is valid. FSS 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 the NotBefore and NotOnOrAfter values represent the beginning and end of the validity interval.

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

**NotBefore=IssueInstant - Skew Time**

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

### Calculating Assertion Validity if FSS is at Both Sides of the Partnership

If FSS is at both sides of a partnership, the total time the assertion is valid 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 FSS, 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 takes the NotBefore and NotOnOrAfter values from the assertion and applies its skew time to those values to calculate new NotBefore and NotOnOrAfter values.

Skew Time = 180 seconds (3 minutes)

NotBefore = 4:56PM

NotOnOrAfter=5:05PM

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

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

## 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 Policy Server User Interface 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 Policy Server User Interface.
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.

**More Information:**

[Restrictions Dialog Field and Controls](#) (see page 606)

## 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 Policy Server User Interface.
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.

**More Information:**

[Restrictions Dialog Field and Controls](#) (see page 606)



## Configure Attributes for WS-Federation Assertions (optional)

Attributes can provide information about a user requesting access to a resource at a Resource Partner. 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:** Attributes statements are not required in an assertion.

Attributes can be used by servlets, Web applications, or other custom applications 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 Resource Partner. For example, you can send an attribute variable called Authorized Amount and set it to a maximum dollar amount 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 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, Federation Security Services 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, Federation Security Services 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.

You configure attributes in the Attributes tab of the Resource Partner Properties dialog box. This involves choosing an Attribute Kind then filling in values for the variable name and attribute value.

## Configure Assertion Attributes for WS-Federation

### To configure assertion attributes

1. Log on to the Policy Server User Interface.
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.

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.

### More Information:

[Resource Partner Properties Dialog--Attributes Tab](#) (see page 607)

## 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, you must select the appropriate radio button in the Attribute Kind group box of the Attribute Setup tab.

## Allow Access to the Federation Web Services Application

After you add affiliates to an affiliate domain, the affiliates need permission to access the Federation Web Services application. When you install the Policy Server and Policy Server Option Pack, the FederationWebServicesDomain is installed by default. This domain includes the following policies:

- FederationWSAssertionRetrieval
- FederationWSNotificationService
- FederationWSSessionServicePolicy
- SAML2FWSArtifactResolutionServicePolicy

### To specify permission to the Federation Web Services application

1. From the Domains tab, expand FederationWebServicesDomain and select Policies.
2. Select one of the policies, and click Edit, Properties of Policy.

For SAML 1.x, you need to permit access to:

- FederationWSAssertionRetrieval
- FederationWSNotificationService
- FederationWSSessionServicePolicy

For SAML 2.0, you need to permit access to  
SAML2FWSArtifactResolutionServicePolicy

The SiteMinder Policy dialog opens.

3. From the Users tab, select one of the following:
  - FederationWSCustomUserStore tab for SAML 1.x
  - SAML2FederationCustomUserStore tab for SAML 2.0.

The Users/Groups dialog opens.

The consumers, Service Providers, and Resource Partners are the "users" included in the listed user stores.

4. Click Add/Remove on the appropriate tab.
5. From the Available Members list, choose the affiliate domains that should have access to Federation Web Services then move them to the Current Members list.
6. Click OK to return to the Policy List.
7. Repeat this procedure for all policies relevant for the SAML version you are using.

## 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

If a user visits the Account Partner before going to the Resource Partner, there needs to be a link the user can select that generates an HTTP Get request to the Account Partner's Single Sign-on Service. The hard-coded link that you create must point to this service and must contain the RP Provider ID and, optionally, parameters, such as the wct parameter, the value of which must contain the time in UTC format.

The syntax for the link to the Single Sign-on Service is as follows:

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

**ap\_server:port**

Specifies the server and port number of the system at the Account Partner that 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

If a user starts at the Resource Partner to initiate single sign-on, typically the user goes to a site selection page at the Resource Partner to choose from a list of Account Partners where authentication takes place. 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 AP and must contain the Provider ID of RP and optionally, other parameters, such as wct, which holds the value of the time in UTC format. After the link is selected, the user's browser is redirected to the Account Partner to get the assertion.

## Configure Signout

Ensuring that a user is completely logged out of all sessions ensures security. Signout is the process of a user being logged out of all of his user sessions associated with the browser that initiated the logout.

Signout is triggered by a user-initiated logout and is implemented using a Signout Servlet at Account Partner and Resource Partner sites. A user initiates a signout request from an Account Partner or a Resource Partner by clicking a link pointing to the respective signout servlet that then triggers the signout process.

Be aware that 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.

**Note:** SiteMinder only supports the WS-Federation Passive Request profile for signout.

By configuring the settings on the Signout tab, you are informing the Account Partner whether the Resource Partner supports signout, and if so, how signout is handled.

If you enable signout, you must also:

- Enable the session server at the Account Partner using the Policy Server Management Console
- Configure persistent sessions for the realm with the protected resources at the Resource Partner

## Enable Signout

### To configure signout

1. Log in to the Policy Server User Interface.
2. Access the Resource Partner Properties dialog for the Resource Partner you want to configure.
3. Select the Signout tab.
4. Select the Enable Signout checkbox.

The URL fields become active.

5. 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://.

6. Click OK.

**More Information:**

[Resource Partner Properties Dialog--Signout Tab](#) (see page 609)

## Validate Signout Requests that are Digitally Signed

By default, signature processing is enabled because it is required by the WS-Federation Passive Requester profile; therefore, it *must* be enabled in a production environment. WS-Federation signout requests are always signed by SiteMinder, but no configuration is required in the Policy Server User Interface. You only 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 on the General tab.

To validate signatures of signout requests, there are configuration steps in the Policy Server User Interface and the smkeydatabase.

**To set-up validation**

1. Add the public key to the Account Partner's smkeydatabase.

The public key must correspond to the private key and certificate that the Resource Partner used to do the signing.

**Note:** To see changes to the smkeydatabase immediately, restart the Policy Server. Otherwise, the database updates based on the frequency you configure in the smkeydatabase.properties file.

2. In the Policy Server User Interface, select the Enable Signout checkbox on the Signout tab.

If you select this check box, the Account Partner will validate the signature of the signout request and response.

**More Information:**

[Manage the Key Database for Signing and Encryption](#) (see page 479)

## Customizing Content in WS-Federation Assertions

The WS-Federation Assertion Generator produces SAML assertions. The assertions are used to authenticate users in a federation 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 box.

**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.

For reference information about the WS-Federation 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*. This information applies to the WS-Federation Assertion Generator as well as the SAML Assertion Generator.

For overview and conceptual information, see the *SiteMinder Programming Guide for Java*.

### More Information:

[Resource Partner Properties Dialog--Advanced Tab](#) (see page 610)



## Integrate the Assertion Generator Plug-in with SiteMinder (SAML 2.0/WS-Federation)

If you write an assertion generator plug-in, you have to integrate the plug-in to work with SiteMinder.

To compile the assertion plug-in Java file, see the instructions in the SAML2AssertionSample.java file in the directory:

sdk/samples/assertiongeneratorplugin

### To integrate the assertion generator plug-in with SiteMinder

1. Compile the assertion plug-in Java file.

This file requires the following .jar files 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 enables the plug-in to be loaded.

**Note:** Do not modify the classpath for xercesImpl.jar, xalan.jar, or SMJavaApi.jar.

3. In the Policy Server User Interface, specify the plug-in that SiteMinder should use. Access the Advanced tab in the Service Provider Properties or Resource Partner Properties dialog and complete the following fields:

#### Full Java Class Name

Specify a Java class name for an existing plug-in

#### Parameter

Specify a string of parameters to be 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 via the Policy Server User Interface, you can use the Policy Management API (C or Perl) to integrate the plug-in. For instructions, 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.

**To enable the Assertion Generator to include attributes from a web application in an assertion**

1. Compile the assertion plug-in Java file.

This file requires the following .jar files installed with the Policy Server:

- *policy\_server\_home/bin/java/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 enables the plug-in to be loaded.

**Note:** Do not modify the classpath for xercesImpl.jar, xalan.jar, or SMJavaApi.jar.

3. Configure a sample plug-in.

There is an APIContext class in the SMJavaAPI that has a new method, getAttrMap(), which returns a map object containing the attributes from the web application to be included in the assertion. In the SiteMinder SDK, there are two sample Assertion Generator plug-ins that show how to use this map object:

- SAML2AppAttrPlugin.java (SAML 2.0)
- WSFedAppAttrPlugin.java (WS-Federation)

These samples are located in the directory sdk/samples/assertiongeneratorplugin. They enable the Assertion Generator to add attributes from a web application to the Assertion Generator for inclusion in an assertion.

4. In the Policy Server User Interface, specify the plug-in you are using. Access the Advanced tab in the Service Provider Properties or Resource Partner Properties dialog and complete the following fields:

**Full Java Class Name**

Specify the Java class name for the plugin, For example, the sample classes included with the SiteMinder SDK are:

- com.ca.assertiongenerator.SAML2AppAttrPlugin  
(SAML 2.0)
- com.ca.assertiongenerator.WSFedAppAttrPlugin  
(WS-Federation)

**Parameter**

Specify a string of parameters to be passed to the plug-in specified in the Full Java Class Name field. These parameters would be the attributes you want to include in the assertion.

**Note:** Instead of specifying the assertion plug-in class and its parameters via the Policy Server User Interface, you can use the Policy Management API (C or Perl) to integrate the plug-in. For instructions, see the *SiteMinder SiteMinder Programming Guide for C* or the *SiteMinder Programming Guide for Java*.

5. Restart the Policy Server.

Restarting the Policy Server ensures that the latest version of the assertion plug-in is picked up after being recompiled.

## Protect the Authentication URL to Generate 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. The `redirect.jsp` file must be protected by a SiteMinder policy 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 user's browser. 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

**To create a policy to protect the Authentication URL**

1. Log into the Policy Server User Interface.
2. From the System tab, create Web Agents to bind to the realms that you will 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 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 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.



# Chapter 17: Configure SiteMinder as a Resource Partner

---

This section contains the following topics:

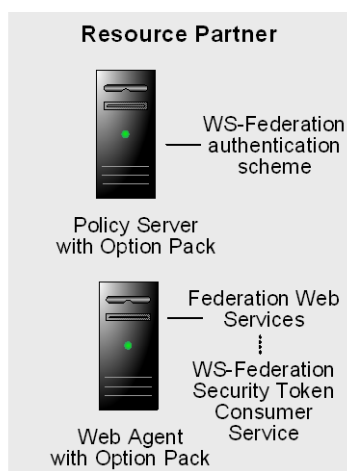
[WS-Federation Authentication Scheme Overview](#) (see page 431)  
[Configuration Tasks for WS-Federation Authentication](#) (see page 433)  
[WS-Federation Authentication Scheme Prerequisites](#) (see page 433)  
[Configure the WS-Federation Authentication Scheme](#) (see page 434)  
[Create a Custom WS-Federation Authentication Scheme](#) (see page 435)  
[Locate User Records for Authentication](#) (see page 436)  
[Configure WS-Federation Single Sign-on at the Resource Partner](#) (see page 438)  
[Implement WS-Federation Signout](#) (see page 440)  
[Customize Assertion Processing with the Message Consumer Plug-in](#) (see page 441)  
[Supply SAML Attributes as HTTP Headers](#) (see page 444)  
[Specify Redirect URLs for Failed WS-Federation Authentication](#) (see page 449)  
[How To Protect a Target Resource with a WS-Federation Authentication Scheme](#) (see page 450)

## WS-Federation Authentication Scheme Overview

If you purchased the Policy Server and Policy Server Option Pack 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 may want to 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 may 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 SiteMinder Federation Web Services application functions. For information about installing and configuring the SPS federation gateway, see the *CA SiteMinder 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's 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.



## Configuration Tasks for WS-Federation Authentication

The following table lists configuration tasks to create a WS-Federation authentication scheme.

| Check Here | Task                                                                                                                                                                                  |
|------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|            | Create an authentication scheme for each Account Partner that generates assertions.                                                                                                   |
|            | Associate that scheme with a realm and add that realm to the policy. You can do this on a per Account Partner basis or create a single custom authentication scheme and single realm. |

### Notes:

- Certain parameter values at the Account Partner and Resource Partner must match for the configuration to work. A list of those parameters can be found in [Configuration Settings that Must Use the Same Values](#) (see page 503).
- To ensure you are using the correct URLs for the Federation Web Services servlets, review the list of URLs, which can be found in [Federation Web Services URLs Used in SiteMinder Configuration](#) (see page 623).

### More Information:

[How To Protect a Target Resource with a WS-Federation Authentication Scheme](#) (see page 450)

## WS-Federation Authentication Scheme Prerequisites

The following are prerequisites for implementing the WS-Federation authentication scheme:

- **Install the Policy Server and Its Option Pack**  
The Policy Server together with the Policy Server Option Pack provides the authentication scheme at the Resource Partner. It also provides the WS-Federation assertion generator used by an Account Partner.  
Install the Policy Server and the Policy Server Option Pack at the producing and consuming sites.  
For installation instructions, refer to the following guides:
  - *SiteMinder Policy Server Installation Guide*
  - *Policy Server, Web Agent Option Pack Release Notes*

- Install either of the following at the Account Partner and Resource Partner sites:
  - Web Agent and the Web Agent Option Pack
  - SPS federation gateway

The Web Agent/Web Agent Option Pack or the SPS federation gateway provide the Federation Web Services application, which enables Resource Partner to consume assertions and supplies other federation services for SiteMinder.

For installation instructions, see the relevant guides:

- *SiteMinder Web Agent Installation Guide*
- *Policy Server, Web Agent Option Pack Release Notes*
- *SiteMinder Secure Proxy Server Administration Guide*
- Set Up the SmKeyDatabase to Sign and Verify Responses

#### More Information

[Manage the Key Database for Signing and Encryption](#) (see page 479)

## 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. Check the [WS-Federation Authentication Scheme Prerequisites](#) (see page 433).
2. Log into the Policy Server User Interface.
3. From the menu bar, select Edit, System Configuration, Create Authentication Scheme.

The Authentication Scheme Properties dialog box opens.

4. From the Authentication Scheme Type drop-down list, select WS-Federation Template.

The contents of the SiteMinder Authentication Scheme dialog box change for the scheme.

5. Configure the scheme common setup group box 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. Ensure the Disable Signature Processing checkbox 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 checking the Disable Signature Processing option.

After configuring an authentication scheme, associate the scheme with a realm that contains the resource you want to protect.

**More Information:**

[Authentication Scheme Properties Dialog--WS-Federation Template](#) (see page 611)

[Federation Authentication Scheme Dialog--Scheme Setup Tab](#) (see page 613)

[How To Protect a Target Resource with a WS-Federation Authentication Scheme](#) (see page 450)

## Create a Custom WS-Federation Authentication Scheme

The Advanced tab of the Authentication Scheme Properties dialog box 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`.

**More Information:**

[Auth. Scheme Properties--WS-Federation--Advanced Tab](#) (see page 620)

## 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 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 value in the assertion; however, you can also obtain the LoginID by specifying 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, if 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 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 is optional.

## Configure Disambiguation Locally

To locate a user record in a local user directory you have to disambiguate the user. Disambiguation is a two-step process:

1. Obtain the LoginID--either by the default behavior of extracting it from the assertion or by using an Xpath query.
2. Using the LoginID, locate the user in the user store--either by the default behavior of passing the LoginID to the Policy Server or using a search specification.

**Note:** The use of Xpath and 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 should not contain namespace prefixes. The following 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.

### More Information:

[Auth. Scheme Properties--WS-Federation--Users Tab](#) (see page 616)

## 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 box 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 checked against the user store to find the correct record for authentication.

5. Click OK to save your configuration changes.

**More Information:**

[Auth. Scheme Properties--WS-Federation--Users Tab](#) (see page 616)

## 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, Federation Security Services 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, Federation Security Services 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.

#### More Information:

[Enforce a Single Use Policy to Enhance Security](#) (see page 439)  
[Auth. Scheme Properties--WS-Federation--SSO Tab](#) (see page 617)

## Enforce a Single Use Policy to Enhance Security

The single use policy feature prevents a WS-Federation assertion from being re-used at a Resource Partner to establish a second session.

Ensuring that an assertion is used only one time is an additional security measure for authenticating across a single sign-on environment. It mitigates security risks caused when an attacker acquires a security token response message from a user's browser that has already been used to establish a SiteMinder session. The attacker can then post the assertion to the WS-Federation Assertion Consumer Service at the Resource Partner to establish a second session.

A single use policy is enabled by a storage mechanism provided by the SiteMinder Session Server. This mechanism is expiry data. Expiry data ensures a single use policy for WS-Federation assertions by storing time-based data about an assertion. The WS-Federation authentication scheme uses the expiry data interface to access the expiry data in the Session Server database.

## How the WS-Federation Single Use Policy is Enforced

Upon successful validation of an assertion, the WS-Federation authentication scheme writes assertion data in the expiry data table with a key of the assertion ID and an expiration time. The Session Server Management thread in the Policy Server deletes expired data from the expiry data table.

If single policy use is enforced, writing assertion data will fail if an entry already exists in the expiry data table with a key of the assertion ID because the assertion has already been used to establish a session. If the scheme cannot write to the table in the session server, the WS-Federation authentication scheme denies the authentication in the same manner as an invalid assertion.

Writing assertion data may fail for other reasons; however, if the single use of the assertion cannot be enforced because the database is unavailable for any reason, then the authentication scheme will deny the request to ensure that assertions cannot be re-used.

### Configure a Single Use Policy

#### To configure a single use policy

1. From the Authentication Scheme Properties dialog box, click Additional Configuration.

The WS-Federation Auth Scheme Properties dialog box opens.

2. Select the SSO tab.

The Enforce Single Use Policy checkbox is selected by default.

3. Enable the session server.

#### More Information:

[Enforce a Single Use Policy to Enhance Security](#) (see page 439)

[Storing User Session, Assertion, and Expiry Data](#) (see page 207)

## Implement WS-Federation Signout

Signout allows near-simultaneous logout of all sessions for browser session associated with a particular user. The benefit of the signout feature is that it ensures no sessions are left open within a browser session for unauthorized users to gain access to resources at the Resource Partner.

Signout can be initiated via user's browser from a link at the Resource Partner or at the Account Provider that directs the browser to the Signout servlet at the Account Partner. This servlet, which is a component of Federation Web Services, processes signout requests and responses coming from a Resource Partner or Account Partner; however, the servlet does not need to know the originator of the request or response. It uses the SiteMinder session cookie to determine which session to end.



## Enable Signout

### To configure signout

1. Log on to the Policy Server User Interface.
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:

[Auth. Scheme Properties--WS-Federation--Signout Tab](#) (see page 619)  
[Storing User Session, Assertion, and Expiry Data](#) (see page 207)

## Customize Assertion Processing with the Message Consumer Plug-in

The Message Consumer Plug-in is SiteMinder's Java program that implements the Message Consumer Extension API. Using this plug-in you can implement your own business logic for processing assertions, such as rejecting an assertion and returning a SiteMinder-defined status code. This additional processing works together with SiteMinder's 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 is configured. The plug-in then has the opportunity to disambiguate the user, if it knows how.

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 should return a `UserNotFound` error, which is documented in the `MessageConsumerPlugin` interface. The plug-in's use of SiteMinder's redirect URLs feature is optional and is based on the error code returned by the plug-in. If the Message Consumer plug-in is not configured, the redirect URLs are used based on the error generated by the SAML authentication scheme.

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, but this is not required.

To integrate the Message Consumer plug-in with SiteMinder, use the parameter values that you specify for the plug-in configuration. The plug-in configuration is part of the SAML 1.x, SAML 2.0 and WS-Federation authentication scheme configuration.

**More information:**

[Specify Redirect URLs for Failed SAML 1.x Authentication](#) (see page 293)

[Specify Redirect URLs for Failed SAML 2.0 Authentication](#) (see page 388)

[Specify Redirect URLs for Failed WS-Federation Authentication](#) (see page 449)

## Configure the Message Consumer Plug-in for WS-Federation

**To customize assertion processing**

1. Log on to the Policy Server User Interface.
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. Implement the plug-in class.

A sample class, `MessageConsumerSAML20.java`, can be found in `sdk/samples/authextensionsaml20`.

**Note:** This sample class is used for both SAML 2.0 and WS-Federation.

6. In the Full Java Class Name field, enter the Java class name of the plug-in. This plug-in is invoked by the Message Consumer at run time.

The plug-in class can parse and modify the assertion, and then return the result to the Message Consumer for final processing.

Only one plug-in is allowed for each authentication scheme. For example, `com.mycompany.messageconsumer.SampleCode`

A sample plug-in is included in the SDK. You can view a sample message consumer plug-in at `sdk/samples/authextensionsaml20`.

**Note:** Specify a Message Consumer plug-in for each authentication scheme.

**More Information:**

[Auth. Scheme Properties--WS-Federation--Advanced Tab](#) (see page 620)

## Integrate the Message Consumer Plug-in with SiteMinder (WS-Federation)

In addition to configuring a message consumer plug-in, you have to integrate the plug-in with the WS-Federation authentication scheme.

The instructions for compiling the message consumer plug-in Java file are in the `AssertionSample.java` file, in `sdk/samples/authextensionsaml20`.

**To integrate the Message Consumer plug-in with the authentication scheme**

1. Compile the message consumer plug-in Java file.

The Java file requires the following .jar file installed with the Policy Server:

`<policy_server_home>/bin/java/SmJavaApi.jar`

2. In the `JVMOptions.txt` file, modify the `-Djava.class.path` value so it is set to the classpath for the plug-in. This enables the plug-in to be loaded.

**Note:** Do not modify the classpath for `SMJavaApi.jar`.

3. Restart the Policy Server.

Restarting ensures that the latest version of the message consumer plug-in is picked up after being recompiled.

**Note:** Instead of specifying the message consumer plug-in class and its parameters via the Policy Server User Interface you can use the Policy Management API (C or Perl). For instructions, see the *SiteMinder Programming Guide for C* or the *SiteMinder Programming Guide for Java*.

Additional information about the Message Consumer 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 *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*.

## Supply SAML Attributes as HTTP Headers

An assertion response may include attributes in the assertion. These attributes can be supplied as HTTP header variables and used by a client application can use these headers for finer grained access control.

The benefits of including attributes in HTTP headers is as follows:

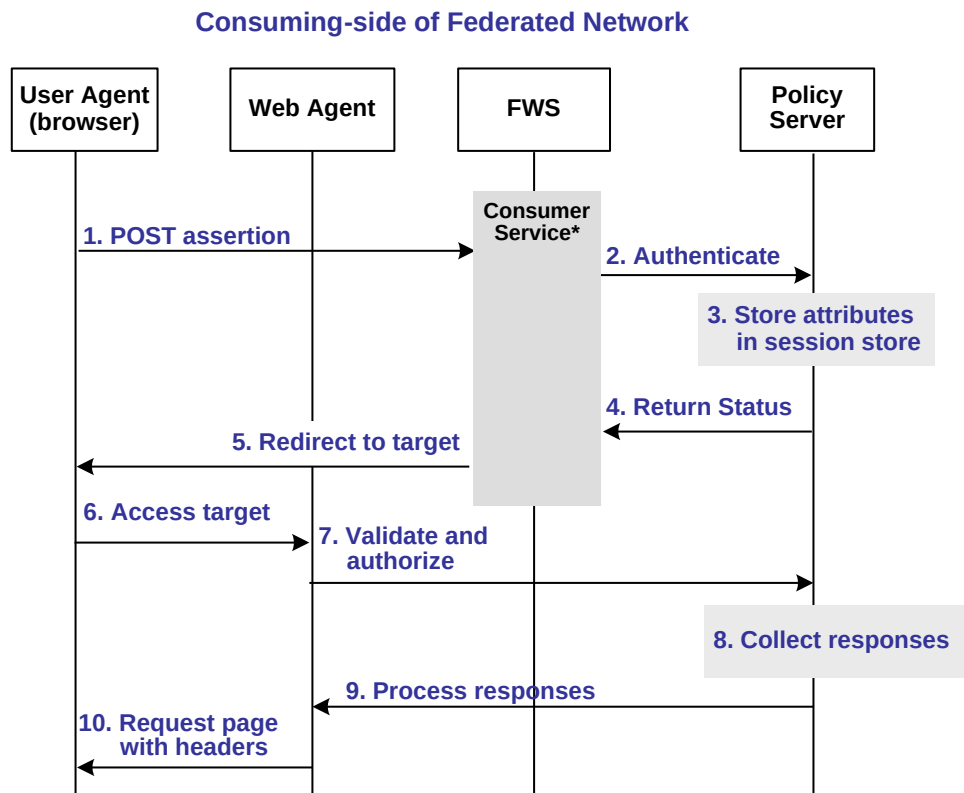
- HTTP headers are not persistent. They are present only within the request or response that contains them.
- HTTP headers, as supplied by the SiteMinder Web Agent, are not seen by the user's browser, which reduces security concerns.

**Note:** The HTTP headers have size restrictions that the attributes cannot exceed. Federation Security Services 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's application.

The following flow diagram shows the sequence of events at runtime:



\*Consumer service can be one of the following:

- SAML Credential Collector (SAML 1.x)
- Assertion Consumer Service (SAML 2.0)
- Security Token Consumer Service (WS-Federation)

**Note:** The SPS federation gateway can replace the Web Agent and Web Agent Option Pack to provide the SiteMinder 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 *CA SiteMinder 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 producing partner, it sends the assertion to the appropriate consumer service at the consuming partner. 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's 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's session and to ensure 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

There are several configuration steps required to retrieve the SAML attributes cached in the session store and provide them as HTTP headers.

### **The components you must configure are as follows**

1. Select `PersistAttributes` as the redirect mode for the SAML authentication scheme. This enables the SAML Attributes to be returned as HTTP headers.
2. Configure an authorization rule for the realm that contains the target resource.
3. Configure a response that uses the active response type for each SAML attribute to be supplied as a header.
4. 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 Policy Server User Interface.
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, you need to 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, therefore the Web Agent cannot re-authenticate the user and pass on the HTTP headers. So, the retrieval of the attributes must happen during the authorization stage.

### To create an OnAccessAccept Rule for the realm

1. Log on to the Policy Server User Interface.
2. From the Domains tab, navigate to the realm which protects the target resource.
3. Select the realm with the target resource and choose 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. Choose 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 group box..
8. Ensure that Enabled is checked in the Allow/Deny and Enable/Disable group box.
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

You must configure a response that sends the SAML attributes as HTTP headers to the Web Agent. The Web Agent will process the response and make the header variables available to the client application.

### To create a response to send the attributes as headers

1. Log on to the Policy Server User Interface.
2. From the Domains tab, select the domain that contains the target resource.
3. In the domain, select the Responses object and create a new 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 group box.
7. Complete the fields in the Attribute Fields group box as follows:

#### Variable Name

Specify the name you want for the header variable. You assign this name.

#### Library Name

smfedattrresponse

This must be the entry for this field.

#### Function Name

getAttributeValue

This 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 will be in the assertion.

8. Click on OK to save the attribute.
9. Repeat the procedure for each attribute that should become an HTTP header variable. You can configure many attributes for a single response.

The response will send 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 Policy Server User Interface.
2. From the Domains tab, select the domain that contains the target resource.
3. In the domain, select the Policies object and create a new policy.  
The Policy Properties dialog opens.
4. Enter a descriptive name in the Name field.
5. Select the users that should 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.

## Specify Redirect URLs for Failed WS-Federation Authentication

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 WS-Federation
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, 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.

**More Information:**

[Auth. Scheme Properties--WS-Federation--Advanced Tab](#) (see page 620)

## How To Protect a Target Resource with a WS-Federation Authentication Scheme

After configuring a WS-Federation authentication scheme, you can use the scheme to protect the realm that holds the target resources requested by users. These resources then need to be protected by a SiteMinder policy.

At the Resource Partner, you must configure a WS-Federation authentication scheme for each Account Partner that generates assertions. The Account Partner is identified in the Account Partner ID field of the Scheme Setup tab. Each scheme must then be bound to a realm, which consists of all the target URLs that comprise the Resource Partner resources.

There are two ways to set-up a realm that contains target URLs:

- You can create a unique realm for each authentication scheme and Account Partner already configured.
- You can configure a single target realm 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 Account Partners simplifies configuration of realms for authentication.

**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 site, an administrator needs to include this response in a link so that this link the user gets redirected to the Resource Partner.

## Assign the WS-Federation Scheme to a Realm

At the Resource Partner, you must configure a WS-Federation authentication scheme for each Account Partner that generates assertions. The Account Partner is identified in the Account Partner ID field of the Scheme Setup tab. Each scheme must then be bound to a realm, which consists of all the target URLs that comprise the Resource Partner resources.

There are two ways to set-up a realm that contains target URLs:

- You can create a unique realm for each authentication scheme and Account Partner already configured.
- You can configure a single target realm 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 Account Partners simplifies configuration of realms for authentication.

**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 site, an administrator needs to include this response in a link so that this link the user gets redirected to the Resource Partner.

## Create a Unique Realm for Each WS-Federation Authentication Scheme

The process for configuring a unique realm for each WS-Federation authentication scheme, follows the standard instructions for creating realms using the Policy Server User Interface.

## 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, set-up:

- A single custom authentication scheme  
You should have already configured a WS-Federation authentication scheme for each Account Partner prior to configuring a custom template.
- A single realm with one target URL

## 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 Policy Server User Interface.
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 WS-Fed Custom Auth Scheme.

5. Complete the following field in the Scheme Common Setup group box:

#### **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 should 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.

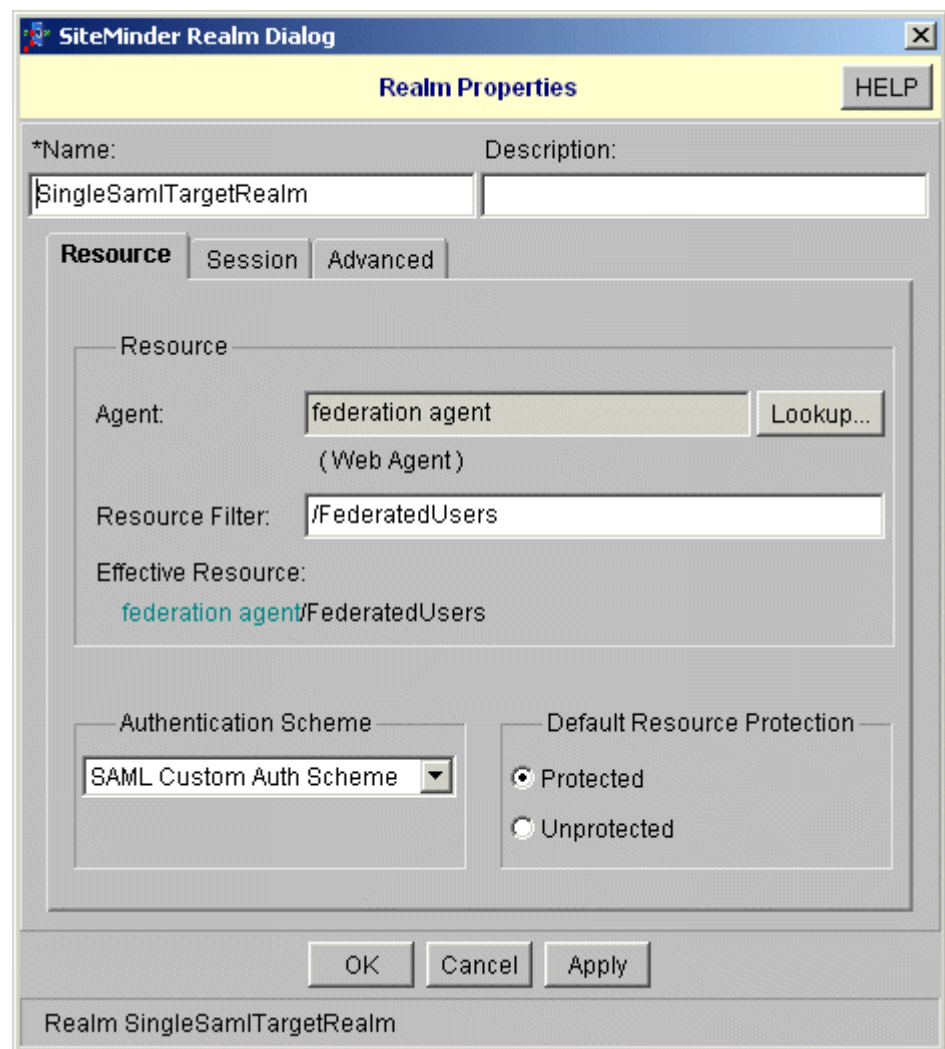
7. Click OK to save your changes.

## Configure a Single Target Realm

#### To create the single target realm

1. Log into the Policy Server User Interface.
2. Display the policy domains.
3. Expand the policy domain where you will add the new realm.
4. Click on the realm icon.
5. From the menu bar, select Edit, Create Realm.  
The SiteMinder Realm dialog box opens.
6. In the Name field, enter a name for this custom target realm.
7. In the Agent field, select a SiteMinder Web Agent protecting the Web server with the target Service Provider resource.
8. In the Resource Filter field, specify the location of the target resource.  
For example, /FederatedUsers
9. From the Authentication Scheme drop-down list, select the custom authentication scheme that you configured for directing requests to the appropriate WS-Federation authentication schemes.
10. Ensure that Protected is selected in the Default Resource Protected group box.
11. Click OK to save the realm configuration.

The following illustration shows the Realm Properties dialog box for the custom realm.



## Create a Rule for the Target WS-Federation Resource

After adding the target resources to a realm, create rules for the realms that protect either the entire contents of the realm or a specific resource in the realm.

## Create a Policy for the WS-Federation Resource

### To protect the target resource

1. From the System tab, create a policy domain.  
Do not create an affiliate domain.
2. Under the policy domain, create a realm that contains the target resources and select the WS-Federation authentication scheme for this realm.
3. Under the realm, create a rule for the resources in the realm and select the Web Agent actions.
4. Create a policy. This policy should contain:
  - users that should have access to the target resources.
  - rule associated with the realm that contains the target resources





# Chapter 18: Use SAML 2.0 Provider Metadata To Simplify Configuration

---

This section contains the following topics:

[SiteMinder SAML 2.0 Metadata Tools Overview](#) (see page 457)

[Export Metadata Tool](#) (see page 458)

[Import Metadata Tool](#) (see page 465)

## SiteMinder SAML 2.0 Metadata Tools Overview

SiteMinder provides a metadata tool to programmatically import and export SAML2 metadata so you can efficiently exchange federation configurations between a site that uses SiteMinder and a partner that may or may not use SiteMinder. Programmatic use of SAML2 metadata should limit the amount of configuration that you perform.

The metadata tool is installed by the Policy Server and Policy Server Option Pack. There are two command-line utilities that make up the SiteMinder metadata tools: 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 may 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

To facilitate federation with sites acting as Service Providers, use the tool to produce a metadata file containing information about profiles supported by the Identity Provider. 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 based on an existing Service Provider

A SiteMinder Identity Provider's generates a metadata file based on an existing Service Provider object defined in the Identity Provider's policy store. The use of the Service Provider object reduces the amount of required data that a user must enter because many of the settings for the Identity Provider metadata file can be derived from the existing Service Provider. Also, the default names of the servlets provided by SiteMinder are used.

Use of the export tool in the way assumes that the Identity Provider's existing relationship with a Service Provider will be similar to the relationship being established, and that the URLs of the servlets for SSO and SLO services are the SiteMinder default servlet names prepended with the IP address and port of the Federation Web Services application, that is,

- `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 an Service Provider metadata file based on an existing SAML 2.0 Authentication Scheme

A SiteMinder Service Provider generates a metadata file based on an existing SAML 2.0 Authentication Scheme object already defined in the Service Provider's policy store. The use of the Service Provider object reduces the amount of required data that a user must enter because many of the settings for the SP metadata file can be derived from the existing SAML 2.0 authentication scheme and the default names of the servlets provided by Siteminder are used.

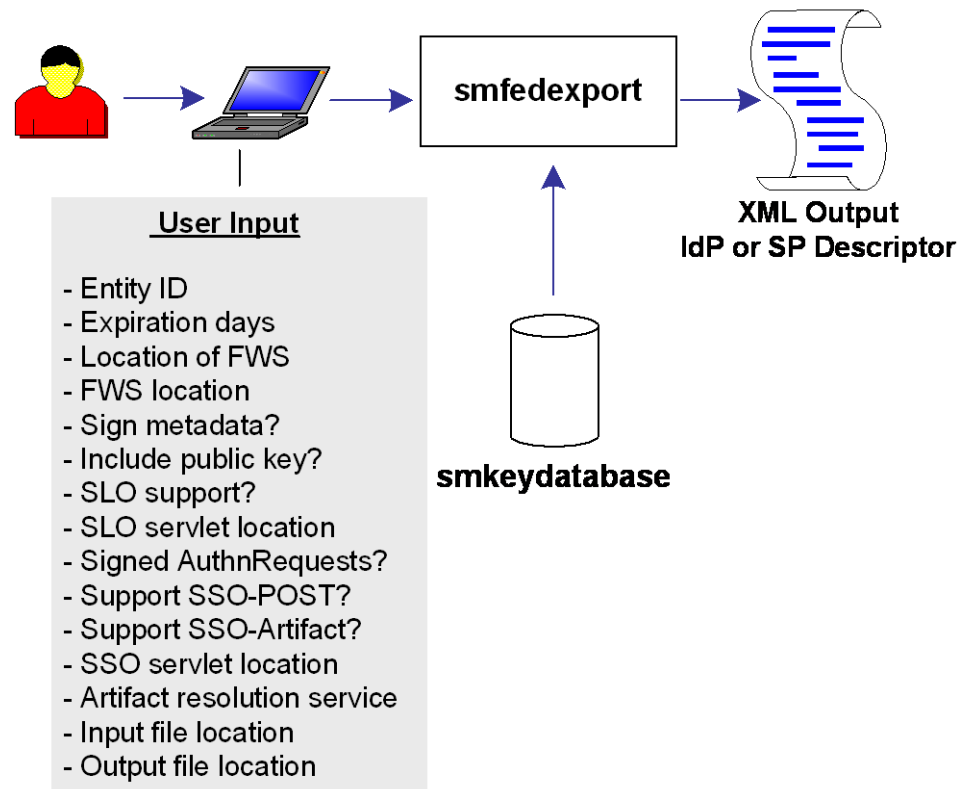
The use of the export tool in this way assumes that the Service Provider's existing relationship with an Identity Provider will be similar to the relationship being established, and that the URLs of the servlets for SSO and SLO services are the Siteminder default servlet names prepended with the IP address and port of the Federation Web Services application, such as,

- `http://idp_server:port/affwebservices/public/saml2sso`
- `http://idp_server:port/affwebservices/public/saml2slo`

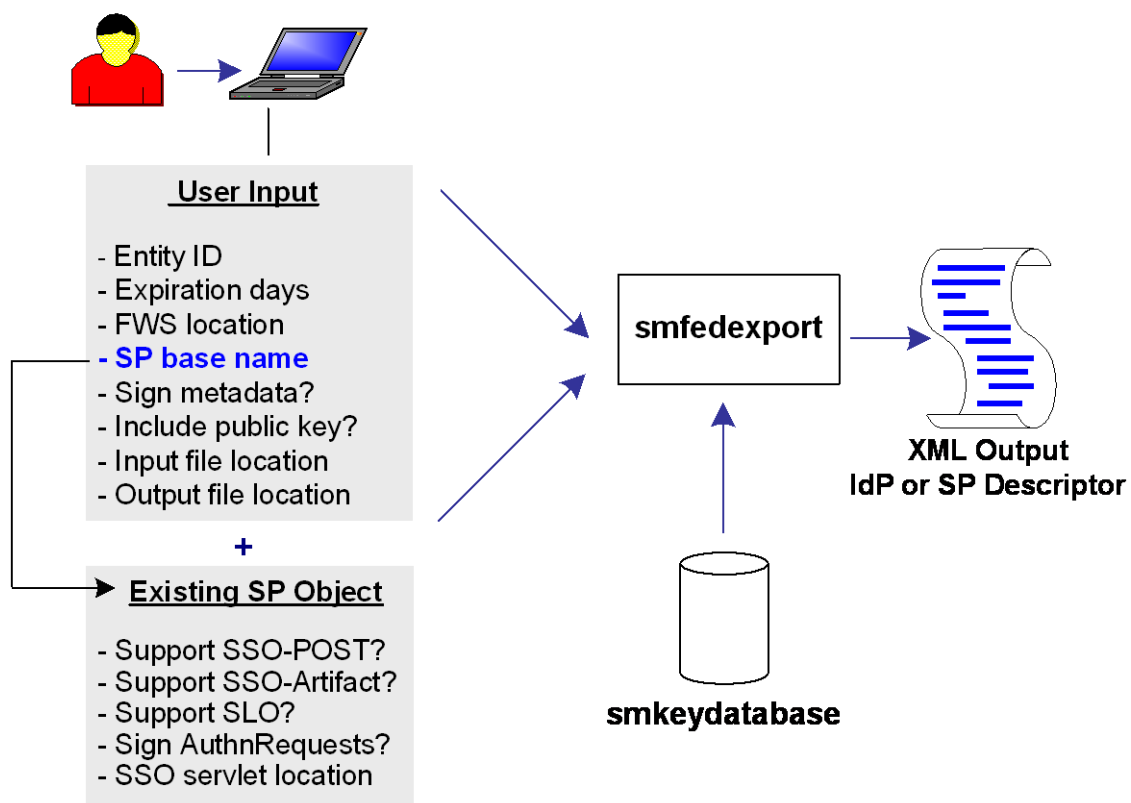
**idp\_server:port**

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

The following figure shows a metadata file generated only from user input.



The following figure 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 machine where you installed the Policy Server and Policy Server Option Pack, open a command window.
2. Enter the smfedexport command using the syntax associated with 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> [-spsbase <spsname>] -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>
-acsisindex <num>][-acsisdef]][-output <file>]
```

#### To sign an existing Metadata document:

```
smfedexport -type (saml2sp|saml2idp) -sign -input <file> -output <file>
```

After running these command, 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 the initial command options are processed by smfedexport, the tool prompts you for additional data based on the type of export file being generated. Any optional arguments not entered on the command-line have defined default values.

**Note:** If you are creating an IdP metadata file, you must have at least one Single Sign-on Service defined in the smfedexport command. If you are creating an SP metadata file, you must have at least one Assertion Consumer Service defined in the smfedexport command.

## Command Options for smfedexport

The smfedexport command-line options are listed in the table that follows:

| Option        | Description                                                                                                                                                                                                                                | Values                                                                                                                                                                       |
|---------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| -acs          | Assertion Consumer Service URL                                                                                                                                                                                                             | URL                                                                                                                                                                          |
| -acsindex     | Assertion Consumer Service index value                                                                                                                                                                                                     | integer                                                                                                                                                                      |
| -acsisdef     | Makes the immediately preceding Assertion Consumer Service the default.                                                                                                                                                                    | none                                                                                                                                                                         |
| -acsbinding   | SAML protocol binding for the Assertion Consumer Service.                                                                                                                                                                                  | <ul style="list-style-type: none"> <li>■ ART (for artifact)</li> <li>■ POST (for POST)</li> <li>■ PAOS (for Reverse SOAP - ECP)</li> </ul>                                   |
| -ars          | Artifact Resolution Service                                                                                                                                                                                                                | URL                                                                                                                                                                          |
| -entityid     | Represents the ID of the SP or IDP whose metadata you are exporting                                                                                                                                                                        | URI                                                                                                                                                                          |
| -expiredays   | Days until the metadata document is no longer valid                                                                                                                                                                                        | integer, 0 is the default<br>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<br><code>http://host:port</code>                                                                                                                             |
| -input        | Full path to an existing XML file                                                                                                                                                                                                          | string, no default                                                                                                                                                           |
| -output       | Full path to an output XML file                                                                                                                                                                                                            | Default values:<br>IDPSSODescriptor.xml<br>SPSSODescriptor.xml                                                                                                               |
| -password     | SiteMinder Administrator name<br>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 does not have to be signed. | true, if present<br>false otherwise                                                                                                                                          |
| -reqsignauthr | Require signed AuthnRequests                                                                                                                                                                                                               | true, if present<br>false otherwise                                                                                                                                          |

| Option            | Description                                                                                                                                                                                                                                                                                                                                              | Values                               |
|-------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------|
| -schemebase       | Points to an existing Service Provider. The settings for the profiles/bindings are taken from this provider.<br>Requires the following options:<br>-fwsurl<br>-username<br>-password                                                                                                                                                                     | authentication scheme name           |
| -spbase           | Points to an existing Service Provider. The settings for the profiles/bindings are taken from this provider.<br>Requires the following options:<br>-fwsurl<br>-username<br>-password                                                                                                                                                                     | Service Provider Name                |
| -sign             | Indicates whether the Policy Server signs the metadata. This setting is optional.                                                                                                                                                                                                                                                                        | true, if present<br>false, otherwise |
| -sigalg           | Designates the signature hashing algorithm SiteMinder uses to for signing assertions and assertion responses, single logout requests and responses                                                                                                                                                                                                       | rsawithsha1<br>rsawithsha256         |
| -signauthr        | Indicates whether the SP signs AuthnRequests                                                                                                                                                                                                                                                                                                             | true, if present<br>false, otherwise |
| -signingcertalias | Specifies the alias associated with the key/certificate pair that signs the metadata. The pair must be stored in the smkeydatabase.<br>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                                  |

| Option              | Description                                                             | Values                                                                                         |
|---------------------|-------------------------------------------------------------------------|------------------------------------------------------------------------------------------------|
| -ssobinding         | SSO Service URL protocol binding                                        | <ul style="list-style-type: none"><li>■ REDIR (for Web SSO)</li><li>■ SOAP (for ECP)</li></ul> |
| -type<br>(Required) | Entity type of the export file                                          | saml2idp<br>saml2sp                                                                            |
| -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 an Service Provider

```
smfedexport -type saml2sp -entityid http://www.myidp.com/sp1
-expiredays 30 -sign -pubkey -slohttpredir http://www.mysite.com/
affwebservices/public/saml2slo -signauthr -aconsvcpst
http://www.mysite.com/affwebservices/public/saml2assertionconsumer
-aconsvcpstindex 12345 -output myidpdescription.xml
```

### Example: Modifying and Signing an Exported Data File

This example assumes that you have already exported metadata to an XML file using the smfedexport tool, but now you want to modify the file and digitally sign it.

#### 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
```

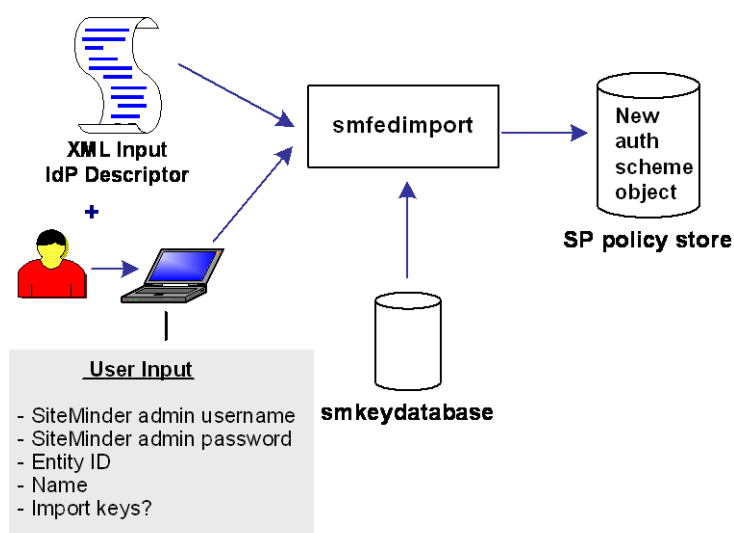
For example:

```
smfedexport -sign -infile myspdescription.xml -output newspdescription.xml
```

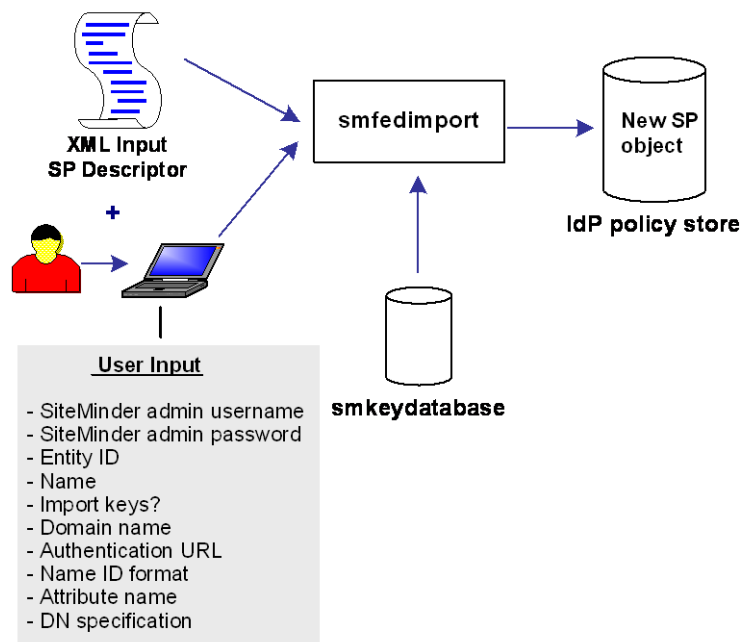
## Import Metadata Tool

You can use the import tool for the following tasks:

- Create a new SAML 2.0 authentication scheme for a Service Provider, as shown in the following figure.



- Create a new 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 will be 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 machine where you installed the Policy Server and Policy Server Option Pack, 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 the initial command options are processed by smfedimport, the tool prompts you for additional, required data based on the type of file that is being imported. Any optional arguments that are not entered on the command-line have defined 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 myaffiliateddomain
-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                                                                                                                                                                                             |
| -entityid     | Entity ID                                                                                                                                                 | The Service Provider ID for the import or the Identity Provider ID for the import                                                                                                                  |
| -importkeys   | Indicates whether or not 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 will be 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, the Identity Provider, or the name of a SAML authentication scheme | string                                                                                                                                                                                             |
| -nameidformat | Name ID format                                                                                                                                            | (U)nspecified--default<br>(E)mail address<br>(X)509 Subject name<br>(W)indows domain name<br>(K)erberos Principal Name<br>E(n)tity Identifier<br>(P)ersistent Identifier<br>(T)ransient Identifier |
| -nameidtype   | Name ID type                                                                                                                                              | (S)tatic<br>(U)ser attribute<br>(D)N attribute                                                                                                                                                     |
| -password     | SiteMinder Administrator password                                                                                                                         | string, no default                                                                                                                                                                                 |

| Option              | Description                                                                                                                                                                                                                                                                                                                                             | Value                               |
|---------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------|
| -type<br>(Required) | Entity type of the import file                                                                                                                                                                                                                                                                                                                          | saml2idp<br>sam2sp                  |
| -silent             | Determines whether the tool interactively prompts the user<br><br>With this option, the tool operates in silent mode. It does not interactively prompt the user for missing input and 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 should be imported. | true, if present<br>false otherwise |
| -username           | SiteMinder Administrator name                                                                                                                                                                                                                                                                                                                           | string, no default                  |

## Processing Import Files with Multiple SAML 2.0 Providers

If there are multiple providers specified in one import file, the tool imports them into the same affiliate domain with names based on the value you specify for the `smfedimport` command option **-name**.

For example, if there are three Service Providers in the import file and you specify:  
`-name mySP`

The tool registers the imported providers as `mysp`, `mysp_1`, and `mysp_2`. The integer increases by one for each subsequent provider. If there is a mixture of Identity Providers and Service Providers in an import file, the naming convention still applies.

## Processing Import Files with Multiple Certificate Aliases

If there are multiple certificates in the import file, the tool imports them into the `smkeydatabase` and 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 19: Federation Security Services

## Trace Logging

---

This section contains the following topics:

[Trace Logging](#) (see page 471)

[Set Up and Enabling Trace Logging](#) (see page 471)

[Simplify Logging with Trace Configuration Templates](#) (see page 475)

## 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 and/or debugging purposes. Trace messages are ordinarily turned off during normal operation, but you can enable them to extract more in-depth information in addition to the trace message itself; for example, the name of the current user or realm. The collected trace messages are written to a trace log.

**Note:** For Web Agents on IIS 6.0 servers, log files are created only after the first user request has been submitted. To check your configuration in the log file, a user has to submit a request.

## Set Up and Enabling Trace Logging

You can establish trace logs at the Web Agent and the Policy Server to monitor SiteMinder operation.

## Log Messages for Federation Web Services at the Web Agent

The Federation Web Services (FWS) application, 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 related activity.
- administration--watches administration-related messages
- request--monitors request and authentication activity.
- general--monitors activity not covered by the other subcomponents.
- configuration--monitors SAML 2.0 Service Provider configuration messages

FWS uses the common tracing facility used by the Web Agent to log trace messages. The following files are used to set up trace logging:

- trace configuration file--the configuration file that determines which components and events FWS monitors. The default file is FWSTrace.conf.
- trace log file--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. It does not define message content.

### Configure FWS Trace Logging

To collect trace messages for the Federation Web Services application, you have to configure the FWS trace logging.

#### To configure FWS trace logging

1. Do one of the following:
  - 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 instructs the trace facility to write messages to a file.
  - Set the TraceFileName parameter to the full path of the trace log file. The default location is in *web\_agent\_home/config/FWSTrace.log*.
  - Set the TraceConfigFile parameter to the full path of the trace configuration file, either the default template, FWSTrace.conf or another template. Templates can be found at *web\_agent\_home/config*.



3. Optionally, you can format the trace log file, the file that contains the log output. The following parameters are the Web Agent configuration parameters that dictate the format of the trace log file:

- TraceRollover
- TraceSize
- TraceCount
- TraceFormat
- TraceDelim

The `LoggerConfig.properties` file contains descriptions of all these settings.

## Log Messages for Federation Services 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.

To configure logging at the Policy Server, use the Policy Server Profiler. The Profiler is available from the Policy Server Management Console, and it is a graphical user interface that lets you specify the following:

- trace configuration file--defines the components and subcomponents that will be included in the file.
- trace log file--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 used for logging. You access it from the Policy Server Management Console.

### To configure the Profiler to collect trace messages for federation services

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 on the Browse button and locate the template you would like to use.

You can load the default template, `smtracedefault.txt`, located in `policy_server_home/config`, or one of the preconfigured templates, located in `policy_server_home/config/profiler_templates`.

5. In the Output group box, select whether the data should be logged 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:** Ensure 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, you need to flush the Federation Web Services cache for the changes to appear in the trace logs.

**Note:** There is a brief delay from when the changes are made and when Federation Web Services receives the information.

### To flush the cache

1. Access the Policy Server User Interface.
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 pre-configured templates are installed with the Policy Server and Policy Server Option Pack 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 Federation Web Services

The following templates are available for Federation Web Services:

| Template           | Tracing Messages Collected                            |
|--------------------|-------------------------------------------------------|
| WebAgentTrace.conf | Default template. Collects data that you specify.     |
| FWS_SSOTrace.conf  | Collects single sign-on messages                      |
| FWS_SLOTrace.conf  | Collects single logout messages                       |
| FWS_IPDTrace.conf  | Collects Identity Provider Discovery Profile messages |

All these templates include the Fed\_Client component and subcomponents related to 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 give it a new name.  
**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 is the file that 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 *SiteMinder Web Agent Guide*.

**Note:** Web Agents installed on IIS 6.0 and Apache 2.0 Web servers do not support dynamic configuration of log parameters set locally in the Agent configuration file. Consequently, when you modify a parameter, the change does not take effect until the Agent is restarted. However, these log settings can be stored and can be updated dynamically if you configure them in an Agent configuration object on the Policy Server.

## Federation Web Services Template Sample

The following is an excerpt from the FWS\_SLOTrace.conf template. The majority 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 will be monitored. It also shows the specific data fields that indicate what each message will contain (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 related to Identity Provider activity |
| samlsp_trace.template  | Collects messages related 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 on the Browse button and locate the template you would like to use.
5. In the Output group box, select whether the data should be logged 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:** Ensure the log file uses a unique name.

6. Click OK to save your changes.

## Service Provider Template Sample

The following 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 what each message will contain 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 information about using the Profiler, see the *Policy Server Management Guide*.



# Chapter 20: Manage the Key Database for Signing and Encryption

---

This section contains the following topics:

[SmKeyDatabase Overview](#) (see page 479)

[Formats Supported by the Smkeydatabase](#) (see page 482)

[What Gets Stored in smkeydatabase?](#) (see page 483)

[Certificate Revocation Lists in the smkeydatabase](#) (see page 484)

[Properties File for the Key Database](#) (see page 485)

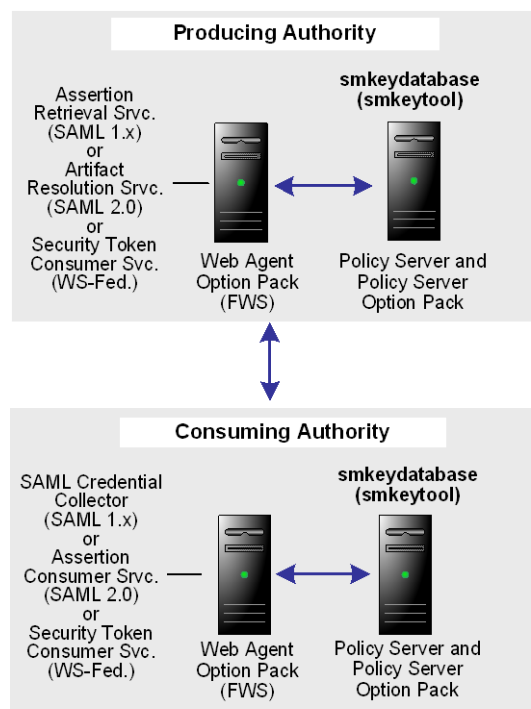
[Modify the Key Database Using smkeytool](#) (see page 486)

[Migrate AM.keystore and Update smkeydatabase](#) (see page 498)

## SmKeyDatabase Overview

The smkeydatabase is a key store used for signing, verification, encryption, and decryption between a SiteMinder consuming authority and a SiteMinder producing authority. This key store resides on the same system where a Policy Server is installed.

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 SiteMinder Federation Web Services application functions. For information about installing and configuring the SPS federation gateway, see the *CA SiteMinder Secure Proxy Server Administration Guide*.

You can store multiple private keys in the smkeydatabase. If you have multiple federated partner, you can use a different private key for each partner.

To create and modify the smkeydatabase, you must use SiteMinder's smkeytool utility.



## Role of the Smkeydatabase at the Producing Authority

At the producing authority, the smkeydatabase is used for the following features:

- Single sign-on (SAML 1.x POST profile, SAML 2.0 POST binding, or WS-Federation)

To use SAML 1.x POST profile, the SAML 2.0 POST binding or WS-Federation Passive Requester Profile for passing assertions, the assertion generator at the producing authority needs to sign the SAML the assertion. The recipient consuming authority needs to verify that signature.

- Encryption of assertions, Name IDs and attributes for SAML 2.0 artifact or POST authentication

If you enable encryption, the producer/Identity Provider must provide the public key certificate of the Service Provider for encrypting the data, while the consumer/Service Provider 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 Consuming Authority

At the consuming authority, 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 consuming authority 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 producing authority, which then returns the assertion to the consuming authority over a back channel.

It is recommended that you protect these services from unauthorized access. To secure the Assertion Retrieval or Artifact Resolution Service, you can 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 consuming authority 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 consuming authority needs to have the Certificate Authority (CA) certificate associated with the server certificate from the producing authority 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 consuming authority must contain the client certificate.

## Aliases in the Smkeydatabase

Aliases enable you to easily reference any single certificate or certificate and private key pair in the smkeydatabase. Every certificate or certificate/private key pair in the smkeydatabase must have a unique alias.

### More Information:

[Migrate AM.keystore and Update smkeydatabase](#) (see page 498)

## 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.

## What Gets Stored in smkeydatabase?

These keys and certificates in smkeydatabase can be at the consuming or producing authority.

The following types of keys and certificates are stored in smkeydatabase:

- Signer's private key and the corresponding certificate, which is the public key that is signed by a certificate authority.

The key and certificate are used to do the following:

- Sign SAML responses for single sign-on, AuthnRequests, and single logout requests
- Decrypt assertions, Name IDs, and attributes.

The smkeydatabase can store multiple private keys and certificates. If a signing alias is configured, the Assertion Generator uses the key associated with that alias to sign assertions. If no signing alias is configured, the Assertion Generator uses the key with the alias **defaultenterpriseprivatekey** to sign assertions. If there is no default enterprise private key found, then the Assertion Generator 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.

- Public-key certificates that correspond to the private keys used to:
  - Sign SAML responses from the assertion issuers
  - Certificates to sign AuthnRequests, certificates used for verifying a single sign-on response, single logout response, or encrypting an assertion, Name ID, or attributes.

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.

## Certificates Stored in the SmkeyDatabase Only at the Consuming Authority

The following types of certificates are stored in smkeydatabase at the consuming authority site:

### **Certificate Authority (CA) certificates**

Used for establishing an SSL connection from a consuming authority to the web server at a producing authority.

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 consuming authority to a producing authority. 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 consuming authority site to ensure the authority 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's 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.

## Properties File for the Key Database

The smkeydatabase properties file, smkeydatabase.properties, defines the configuration parameters required to access and manage the key database.

The smkeydatabase.properties file is installed in:

- *policy\_server\_home*\config\properties (Windows)
- *policy\_server\_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
- NativeDBName
- XMLDocumentOpsImplementation
- AffiliateIXMLSignatureImplementation
- IXMLSignatureImplementation
- EncryptedPassword
- IXMLEncryptDecryptImplementation
- DBUpdateFrequencyMinutes

Descriptions of each setting follow.

## 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 and Policy Server Option Pack 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 consuming authority 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
- addPrivateKey
- addCertOption
- addRevocationInfo
- changepassword
- deleteRevocationInfo
- deleteDB
- delete
- export
- findAlias
- importDefaultCACerts
- listCerts
- listRevocationInfo
- printCert
- renameAlias
- validateCert
- help

A description of each command option follows.

**More Information:**

[Smkeytool Command Options](#) (see page 489)



## Smkeytool Command Options

The smkeytool utility lets you modify the smkeydatabase. Be aware of the following when making database changes:

- If you are adding a private key or 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 automatically update depends on the configured frequency of database updates. You can configure database updates by adjusting the DBUpdateFrequencyMinutes parameter in the smkeydatabase.properties file.

The smkeytool command options and arguments are as follows:

### **-createDB**

Creates a new key database to store keys and certificates. By default, the directory is named smkeydatabase. Additionally, an empty alias store is created and named keyaliases.ser file in the smkeydatabase directory. You can change the key store location by modifying the smkeydatabase.properties file.

**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.

Options 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 Policy Server Option Pack and contains all default CA certificates. This option is the same as executing the -importDefaultCACerts option.

### **-addPrivKey**

Adds the specified private keys and corresponding certificates to the key database. You can have multiple private keys and certificates in the database. Only RSA keys are supported. When you use the -addPrivKey command, you can specify the key data by combining the -keyfile and -certfile options or by using the -keycertfile option alone.

The Policy Server at the producing authority uses a single enterprise private key to sign SAML messages and to decrypt encrypted SAML messages received from the consuming authority. Typically, the enterprise key is the first private key found in the smkeydatabase.

**Note:** The entire smkeydatabase is encrypted; however, the individual private keys are not.

Options for -addPrivKey are as follows:

#### **-alias <alias>**

Required. Alias associated with a single certificate in the database. Must be a unique string and should contain only alphanumeric characters.

#### **-certfile <cert\_file>**

Full path to the location of the certificate associated with this private key. Required for keys in PKCS1, PKCS5, and PKCS8 format.

#### **-keyfile <private\_key\_file>**

Full path to the location of the the private key file. Required for keys in PKCS1, PKCS5, and PKCS8 format.

#### **-keycertfile <key\_cert\_file>**

Full path to the location of the PKCS12 file that contains the private key and public certificate data. Required for keys in PKCS12 format.

#### **-password <password>**

Optional. Private key files are typically encrypted prior to be added to the smkeydatabase. When added to the smkeydatabase, the key needs to be decrypted. The password value represents the password used to decrypt the private key data. This password is not stored in the smkeydatabase.

### **-addCert**

Adds a certificate to the key database. V1, V2, and V3 versions for X.509 certificate format are supported. DER and PEM encoding formats are supported.

**Note:** For the Policy Server to recognize the new certificate immediately, restart the Policy Server. Otherwise, the database updates based on the frequency you configure in the smkeydatabase.properties file.

If you indicate that you want to trust the certificate as a Certificate Authority, this certificate is always treated as a CA certificate.

Options 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.

**-trustcacert**

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**

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.

Options 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"

**-deleteRevocationInfo**

Deletes a CRL from the database.

Options 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**

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.

Option for -deleteDB is as follows:

**-noprompt**

Optional. The user will not be prompted to confirm the deletion of the database.

**-delete**

Deletes an existing certificate from the smkeydatabase. If the certificate has an associated private key, the key is also deleted.

Option 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**

Exports an existing certificate or a private key from the smkeydatabase. Certificate data is exported using PEM encoding. Private key data is exported using DER encoded PKCS8 format.

Options for -export are as follows:

**-alias <alias>**

Required. Identifies the certificate/key to be exported.

**-outfile <out\_file>**

Required. Full path to the output certificate/key file.

**-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 when exporting a private key. Although the entire smkeydatabase is encrypted, individual private keys are stored in unencrypted form. This password encrypts the private key before it is exported.

**-importDefaultCACerts**

Imports all default trusted Certificate Authority certificates from the cacerts.keystore file, which is installed with the Policy Server and Policy Server Option Pack, into the smkeydatabase. Certificate Authority certificates are used to verify the server certificate associated with the producing authority's web server.

**Note:** For the Policy Server to recognize the updates to the smkeydatabase immediately, restart the Policy Server. Otherwise, the database updates based on the frequency you configure in the smkeydatabase.properties file.

**-findAlias**

Determines the alias associated with a certificate that is already in the smkeydatabase.

Option 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.

**-listCerts**

Lists some metadata of all the certificates stored in key database.

Option 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**

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.

Option 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**

Displays some metadata of the specified certificate. This command is especially useful for UNIX systems, where it is difficult to see the certificate properties.

Options 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**

Renames an existing alias associated with a certificate.

Options 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**

Optional. Indicates whether a certificate is revoked or not.

Option for -validateCert is 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**

Shows how to use the smkeytool utility.

## 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 consuming authority 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.



## 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 consuming authority database by entering:  
`smkeytool.sh -listCerts`
2. Add the CA certificate by entering:  
`smkeytool.bat -addCert "c:\program files\ca\siteminder\certs\sampleCARoot.crt" -trustcacert`
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.

## 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 consuming authority. The AM.keystore held Certificate Authority (CA) certificates and client certificates.
- **smkeydatabase**  
This store resided on the producing and consuming 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 consuming authority. Additionally, all certificates must have aliases.

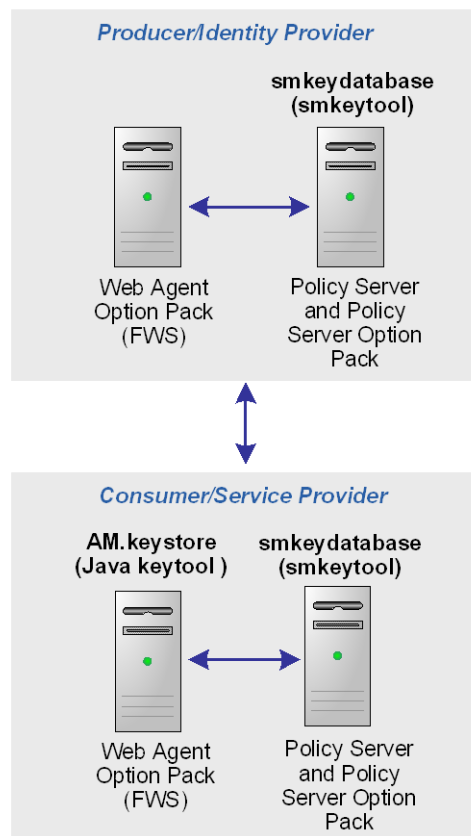
If you are upgrading from versions prior to 6.0 SP 5/6.x QMR 5, you *must* do the following:

1. Copy private keys and certificates from the AM.keystore to the smkeydatabase on the consuming authority's Policy Server.
2. Migrate an existing smkeydatabase so aliases can be added to the certificates in the database.

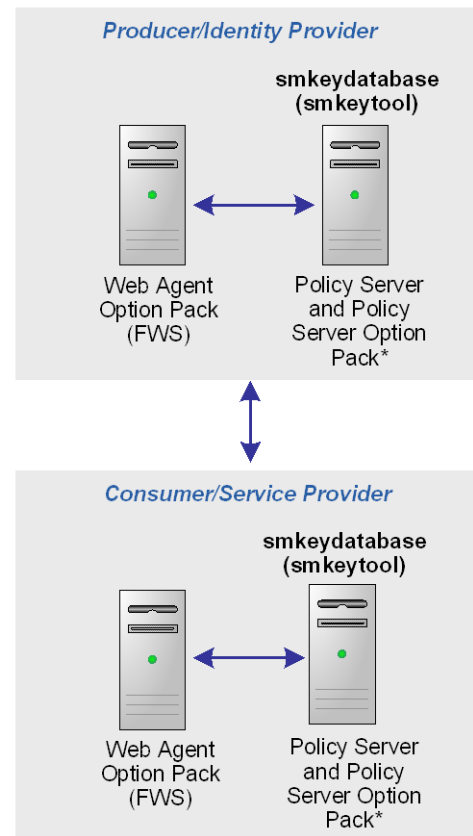
The migratekeystore utility enables you to perform these tasks.

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 501)

## 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 producing authority uses an enterprise private key to sign SAML messages and to decrypt encrypted SAML messages received from the consuming authority.

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 501)

## 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 and Policy Server Option Pack 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 21: Configuration Settings that Must Use the Same Values

---

This section contains the following topics:

[How to Use the Configuration Settings Tables](#) (see page 503)

[SAML 1.x Matching Configuration Settings](#) (see page 503)

[SAML 2.0 Matching Configuration Settings](#) (see page 505)

[WS-Federation Configuration Settings](#) (see page 506)

## 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 be matched with the corresponding value or values described in the other cell(s) in the row.

**Note:** The information is only applicable in an all-SiteMinder environment. That is, where both the producing authority and the consuming authority 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 box or file where these settings are located. Most of these settings are in the Policy Server User Interface; 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...                                                                                                                                |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>&lt;AffiliateName&gt;</b> AffiliateConfig.xml file for SAML Affiliate Agent<br>OR<br><b>Affiliate Name</b> field<br>Scheme Setup tab of the Authentication Scheme Properties dialog (Artifact and POST profiles)             | <b>Name</b> field<br>Affiliate Properties dialog; value must be lowercase<br><b>NAME</b> query parameter in intersite transfer URL links at the producer.                            |
| <b>Verify Password</b> field<br>(SAML Artifact auth. scheme only)<br>Scheme Setup tab of the Authentication Scheme Properties dialog                                                                                            | <b>Confirm Password</b> field<br>Affiliate Properties dialog                                                                                                                         |
| <b>&lt;AssertionAudience&gt;</b> setting<br>AffiliateConfig.xml file; SAML Affiliate Agent is the consumer<br><b>Audience</b> field<br>any other SAML consumer; Scheme Setup tab of the Authentication Scheme Properties dialog | <b>Audience</b> field<br>Assertions tab of the Affiliate Properties dialog                                                                                                           |
| <b>Assertion Consumer URL</b> field<br>(SAML POST auth. scheme only) Scheme Setup tab of the Authentication Scheme Properties dialog                                                                                            | <b>Assertion Consumer URL</b> field<br>Assertions tab of the Affiliate Properties dialog<br><br><b>SMCONSUMERURL</b> query parameter<br>intersite transfer URL links at the producer |
| <b>Issuer</b> field<br>Scheme Setup tab--Authentication Scheme Properties dialog                                                                                                                                                | <b>AssertionIssuerID</b> parameter<br>AMAssertionGenerator.properties file at the producer                                                                                           |
| <b>Version from SAML Version drop down list</b><br>(SAML Artifact auth. scheme only)                                                                                                                                            | <b>Version from SAML Version drop down list</b><br>Assertions tab--Affiliate Properties dialog                                                                                       |
| <b>Company Source ID</b> field<br>(SAML Artifact auth. scheme only)                                                                                                                                                             | <b>SourceID</b> parameter<br>AMAssertionGenerator.properties file at the producer                                                                                                    |



## 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 box or file where these settings are located. Most of these settings are in the Policy Server User Interface; 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...                                                                                                       |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Attribute Name</b><br>Add/Edit Attribute dialog accessed from the Attributes tab of the SAML 2.0 Auth. Scheme Properties dialog                                                                                                                                                                      | <b>Variable Name</b><br>Attribute Fields group box--SAML Service Provider Attribute dialog                                                                  |
| <b>Audience field</b><br>any other SAML Service Provider; SSO tab of the SAML 2.0 Auth Scheme Properties dialog                                                                                                                                                                                         | <b>Audience field</b><br>SSO Tab--SAML Service Provider dialog                                                                                              |
| <b>IdP ID field</b><br>Scheme Setup tab--Authentication Scheme Properties dialog                                                                                                                                                                                                                        | <b>IdP ID field</b><br>General tab--Service Provider dialog<br>For Identity Provider-initiated SSO-- <b>SPID</b> query parameter in an unsolicited response |
| <b>Local Name field</b><br>Add/Edit Attribute dialog accessed from the Attributes tab of the SAML 2.0 Auth. Scheme Properties dialog<br><br><b>Local Name</b><br>Federation Attribute Variable Properties dialog for creating a Federation Attribute variable at the SAML Requester (Service Provider). | None                                                                                                                                                        |
| <b>SP ID field</b><br>Scheme Setup tab--Authentication Scheme Properties dialog<br><br>For Service Provider-initiated SSO-- <b>ProviderID</b> query parameter in hard-coded links to the Identity Provider                                                                                              | <b>SP ID field</b><br>General tab--Service Provider dialog                                                                                                  |

| These Settings at the Service Provider...                                                                                 | Must Match These Settings at the Identity Provider...                            |
|---------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------|
| <b>SP Name</b> field<br>Backchannel tab of the SAML 2.0 Auth Scheme Properties dialog<br>This value must be in lowercase. | <b>Name</b> field<br>Service Provider dialog<br>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 Policy Server User Interface at the Resource Partner.
- The second column, "Setting at the Account Partner", describes a setting that must be configured in the Policy Server User Interface 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...                                                                                                                                                                  |
|-----------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Resource Partner ID</b><br>Scheme Setup tab--Authentication Scheme Properties dialog | <b>Resource Partner ID</b> field<br>General tab Resource Partner Properties dialog<br>wtrealm query parameter should be set to Resource Partner ID for the hard-coded link to trigger Account Partner-initiated SSO. |

| These Settings at the Resource Partner...                                              | Must Match These Settings at the Account Partner...                                      |
|----------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------|
| <b>Account Partner ID</b><br>Scheme Setup tab--Authentication Scheme Properties dialog | <b>Account Partner ID</b> field<br>General tab of the Resource Partner Properties dialog |



# Chapter 22: SAML 1.x Affiliate Dialog Reference

---

This section contains the following topics:

- [Affiliate Dialog](#) (see page 509)
- [Affiliate Dialog Prerequisites](#) (see page 509)
- [Navigate to the Affiliates Dialog](#) (see page 509)
- [Affiliate Dialog Fields and Controls](#) (see page 510)
- [Affiliate Dialog--Users Tab](#) (see page 513)
- [Affiliate Dialog--Assertions Tab](#) (see page 514)
- [Affiliate Dialog--Session Tab](#) (see page 516)
- [Affiliate Dialog--Attributes Tab](#) (see page 516)
- [Affiliate Dialog--IP Addresses Tab](#) (see page 520)
- [Affiliate Dialog--Time Tab](#) (see page 520)
- [Affiliate Dialog--Advanced Tab](#) (see page 521)

## Affiliate Dialog

The Affiliate dialog is where you configure a SAML 1.x consumers.

## Affiliate Dialog Prerequisites

To successfully create a consumer using the Affiliate Dialog, your SiteMinder administrator account must have the Manage System and Domain Objects privilege or the Manage Domain Objects privilege for the affiliate domain in which the consumer is to be added.

## Navigate to the Affiliates Dialog

You can access the Affiliate Properties dialog or Affiliate dialog in the Policy Server User Interface.

**Note:** When you create or modify a Policy Server object in the Policy Server User Interface, use ASCII characters. Object creation or modification with non-ASCII characters is not supported.

**To access the Affiliate Properties dialog**

1. If necessary, in the Object pane, click on the Domains tab.
2. In the Domains tab, click the affiliate domain to which you want to add a consumer.
3. From the menu bar, select:  
Edit *affiliate\_domain\_name*, Create Affiliate

**To access the Affiliate Dialog to edit an existing consumer**

1. If necessary, click on the Domains tab.
2. Expand the affiliate domain that contains the consumer that you want to edit.
3. Click on the Affiliates entry.
4. In the List pane, click the consumer that you want to edit.
5. From the menu bar, select:  
Edit, Properties of Affiliate

## Affiliate Dialog Fields and Controls

The following fields and controls are on the Affiliate dialog:

**Name**

Defines the name of the consumer. This name should be unique across all affiliate domains.

**Description**

Defines a brief description for a consumer.

**Password**

Defines the password that a consumer uses to identify itself to the producer site so it can retrieve an assertion.

**Confirm Password**

Confirms the password entered in the Password field.

**Enabled check box**

If set, activates the consumer. This check box must be marked for the Policy Server and Federation Web Services to authenticate users trying to access consumer resources.

### Use Secure URL

The Use Secure URL feature instructs SiteMinder's Single Sign-on Service to encrypt only the SMPORTALURL query parameter that it appends to the Authentication URL prior to redirecting the user to establish a SiteMinder session. After the user is authenticated, the Authentication URL redirects the user back to the destination specified by the SMPORTALURL query parameter.

Encrypting the SMPORTALURL prevents a malicious user from modifying the value and redirecting authenticated users to a malicious web site instead of the originally requested target.

If you select this checkbox, you must do the following:

1. Set the Authentication URL field to the following URL:

`http(s)://idp_server:port/affwebservices/secure/securedirect`

2. Protect the securedirect web service with a SiteMinder policy.

If the SiteMinder Producer/Identity Provider/Account Partner serves more than one Consumer/Service Provider/Resource Partner, it probably authenticates different users for these different partners. As a result, for each Authentication URL that uses securedirect, this web service must be included in a different realm for each partner.

To associate the securedirect service with different realms, modify the web.xml file to create different resource mappings; you cannot copy the securedirect web service to different locations on your server. This file is located in the directory web\_agent\_home/affwebservices/WEB-INF.

### Allow Notification

If set, SiteMinder provides event notification services for a site where the SAML Affiliate Agent is acting as the consumer. When set, SiteMinder can receive event notifications from the consumer about which affiliate resources a user has accessed.

**Note:** The Notification service is not supported for consumer at which the SAML credential collector acting as a consumer.

### Authentication URL

Defines the protected URL used to authenticate users who do not have a session at the producer.

When a user who has not logged in at the producer requests a protected consumer resource, the user is sent to the 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. This redirect.jsp file is included with the Web Agent Option Pack and with the SPS federation gateway.

**Important!** You must create a policy to protect the AuthenticationURL.

The Affiliates dialog also contains the following tabs:

### Users tab

Lists the users or groups of users for the consumer. Users included in a consumer can be authenticated for access to consumer resources, and the assertion generator can create SAML assertions that include attribute information for the users.

### Assertions tab

Specifies the audience, validity duration, and skew time of a SAML assertion generated by the SAML assertion generator.

### Session tab

Enables the sharing of session information between the producer and the consumer.

### Attributes tab

Configures consumer attributes, which pass user attributes, DN attributes, or static data from the Policy Server to the consumer in an assertion.

### IP addresses tab

Configures IP addresses, ranges of IP addresses, or subnet masks that users must use in order to access a consumer site. If IP addresses have been specified for a consumer, only users who access the consumer site from the appropriate IP addresses will be accepted by the consumer site as producer users.



**Time tab**

Sets time restrictions for an consumer. When you add a time restriction, the consumer only functions during the period specified in the time restriction. If a user attempts to access an consumer resource outside of the period specified by the time restriction, the consumer does not produce SAML assertions.

**Advanced tab**

Optionally, configures a plug-in to customize the content of the SAML assertion generated by the Assertion Generator.

**More Information:**

[Protect the Authentication URL to Create a SiteMinder Session \(SAML 1.x\)](#) (see page 265)

## Affiliate Dialog--Users Tab

The Users tab is where you specify the users and groups that can access resources at the consumer. You can only add users and groups from directories that are included in the affiliate domain in which the consumer exists.

After a user is associated with an consumer object, the assertion generator can create SAML assertions for the user. Then the assertion acts as the user's credentials for access to consumer resources.

**User Directory**

Lists the users and groups that are allowed access resources at the consumer site.

**Add/Remove**

Opens the Users/Groups dialog from where you can add and remove users and groups.

You may only add users and groups from directories that were included in the affiliate domain.

**More Information:**

[Select Users for Which Assertions Will Be Generated](#) (see page 247)

## Affiliate Dialog--Assertions Tab

The Assertions tab is where you define the SAML assertion generated by the Assertion Generator. The dialog contains the following fields:

### SAML Profile

Specifies the SAML profile used for sending an assertion. The options are:

#### Artifact

The artifact is a 42-byte hex encoded ID that references an assertion stored in the session server on the producer-side Policy Server. An artifact lets a consumer retrieve an assertion document from a producer.

#### POST

The POST profile embeds a SAML response with the assertion in an HTML form. The form is posted by the user's browser at the destination consumer site. SAML POST profile is only supported for SAML version 1.1.

### SAML Version

Specifies the version of the SAML protocol in use. The choices are SAML 1.0 and 1.1. SAML artifact can support either version. SAML POST can only support SAML 1.1.

### Assertion Consumer URL

(Required for SAML POST binding, optional for SAML Artifact binding)

Specifies the destination site URL to which the user's browser must send a generated assertion. The default URL varies depending on the SAML binding and whether SiteMinder or the SAML Affiliate Agent is at the consumer site.

- SAML 1.x URL when SiteMinder is the affiliate at the consumer site, the URL is typically the SAML credential collector, which is a component of the Federation Web Services application at the consumer.

`https://consumer_server:port/affwebservices/public/samlcc`

- SAML 1.0 URL when the SAML Affiliate Agent is at the consumer site, set this field to the SSLInterceptorURL that is specified for the SAML Affiliate Agent Configuration. Example:

`https://consumer_server:port/affagent/affiliatesite/test1.htm`

#### consumer\_server:port

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

**Note:** For SAML 1.x artifact binding, the value of the Assertion Consumer URL field takes precedence over the value of the SMCONSUMERURL query parameter, which must be part of the intersite transfer URL that a user selects to initiate single sign-on.

**Audience**

Defines the URL of the document that describes the terms and conditions of the agreement between the producer and the consumer. This value is included in the assertion passed to the consumer and may be used for validation purposes. Also, the consumer may parse the actual audience document to obtain relevant information.

If the SAML Affiliate Agent is the consumer, the value entered here must match the value entered for the Assertion Audience setting in the AffiliateConfig.xml file, the configuration file for the SAML Affiliate Agent. For any other SAML consumer, the value entered must match that of the Audience field configured for the SAML authentication scheme.

**Validity Duration**

Defines the amount of time, in seconds, that the assertion is valid. If the consumer does not receive the assertion during the assertion's valid window, the consumer considers the assertion invalid.

**Skew Time**

Defines the difference, in seconds, between the system clock time of the producer and the system clock time of the consumer. The skew time is added to the validity duration.

**Note:** Times are relative to GMT.

**Sign Assertion**

Enables the producer to digitally sign the assertion with its private key. This adds an additional level of security to the assertion in addition to the assertion response being passed across a secure back-channel.

**Signing Alias**

(Optional) Specifies the alias associated with a specific private key in the SiteMinder key database. By completing this field, you are indicating which private key the Producer should use to sign assertions or assertion responses.

**Note:** It is recommended that the private key already be stored in the key database before you specify its associated alias in this field.

**Limits:** an alphanumeric character string

**More Information:**

[Configure SAML 1.x Assertions to Authenticate Users](#) (see page 250)

## Affiliate Dialog--Session Tab

The Session tab is where you enable sharing of session information between the producer and the SAML Affiliate Agent consumer.

**Note:** Sessions only apply when the SAML Affiliate Agent is the consumer.

The controls on the Session tab are as follows:

### Shared Sessioning check box

If an consumer implements a shared session solution, this check box enables the sharing of session information between the producer and the SAML Affiliate Agent.

### Sync Interval

Defines the frequency (while the user is active at the consumer) at which the SAML Affiliate Agent contacts the producer to validate session status in seconds.

### More Information:

[Set Up Sessions for a SAML Affiliate Agent Consumer \(optional\)](#) (see page 257)

[Set the Sync Interval for Shared Sessions](#) (see page 258)

## Affiliate Dialog--Attributes Tab

The Attributes tab is where you configure whether attributes user attributes, DN attributes, or static data are passed from the producer to the consumer in an assertion. Attributes can be used by servlets, Web applications, or other custom applications to display customized content or enable other custom features.

When used with Web applications, attributes can be used for fine-grained access control by limiting what a user can do at the consumer. For example, you can send an attribute called Authorized Amount and set it to a maximum dollar amount that the user can spend at the consumer.

### Attribute List

Lists the available attributes.

### Create

Opens the Affiliate Attribute Editor dialog from where you configure attributes.

### More Information:

[Configure Attributes to Include in SAML 1.x Assertions \(Optional\)](#) (see page 259)

[Affiliate Attribute Editor Dialog Fields and Controls](#) (see page 517)

## Affiliate Attribute Editor Dialog

To configure attributes to include in SAML assertions for federated communication, you configure the specific attribute name/value pairs in the Affiliate Attribute Editor dialog.

### More Information:

[Configure Attributes to Include in SAML 1.x Assertions \(Optional\)](#) (see page 259)

## Affiliate Attribute Editor Prerequisites

To successfully edit an attribute using the Affiliate Attribute Editor dialog, you must be in the process of creating or editing an affiliate object using the Affiliate dialog.

## Navigate to the Affiliate Attribute Editor Dialog

Use one of the following methods to access the Affiliate Attribute Editor Dialog:

### To create a new attribute

1. Begin at the Affiliate dialog.
2. Click Create on the Attributes tab.

To edit an existing attribute

1. Begin at the Attributes tab.
2. Select an attribute in the Attribute List
3. Click the Edit button.

## Affiliate Attribute Editor Dialog Fields and Controls

Attribute drop down list--Specifies whether the attribute is passed to the consumer as an Affiliate-HTTP-Header-Variable or Affiliate-HTTP-Cookie-Variable response attribute.

## Affiliate Attribute Editor Dialog--Attribute Setup Tab

The Attribute Setup tab of the Attribute Editor dialog contains the following fields:

### Attribute Kind Group Box

The Attribute Kind group box contains radio buttons that specify the attribute type. The fields on this dialog are as follows:

#### Static radio button

If selected, the attribute returns the static value defined in the Variable Value (header variable) or Cookie Value (cookie variable) field.

#### User Attribute radio button

If selected, the attribute returns the information stored in the user attribute specified in the Attribute Name field from a user's entry in a user directory.

#### DN Attribute radio button

If selected, the attribute returns profile information from a directory object associated with a DN in an LDAP or ODBC user directory.

**Note:** 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.

#### Allow Nested Groups check box

If set, nested groups are allowed when selecting the DN Attribute type. Enabled if the DN Attribute radio button is selected.

### Attribute Fields Group Box

The Attribute Fields group box contains fields that specify information about the selected attribute type. The fields in this group box are context-sensitive and are determined according to the selection made in the Attribute Kind group box.

The fields are as follows:

#### Variable Name

Defines the name of the attribute SiteMinder returns to the consumer in a header variable.

#### Cookie Name

Defines the name of the attribute SiteMinder returns to the consumer in a cookie.

#### Variable Value

Defines static text for the value of the attribute name/value pair SiteMinder returns to the consumer in a header variable.

**Cookie Value**

Defines static text for the value of the attribute name/value pair SiteMinder returns to the consumer in a cookie.

**Attribute Name**

Defines the attribute in a user directory that provides the attribute value.

**DN Spec**

Defines the distinguished name of the user group from which SiteMinder should retrieve the user attribute. The DN must be related to the users for whom you want to return values to the consumer.

**Lookup button**

Opens the SiteMinder User Lookup dialog to locate the user group and select a DN to populate the DN Spec field.

**More Information**

[Configure Attributes to Include in SAML 1.x Assertions \(Optional\)](#) (see page 259)

## Affiliate Attribute Editor Dialog--Advanced Tab

The Advanced tab of the attribute editor contains the following field:

**Script**

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 this field for another response.

**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.

## Affiliate Dialog--IP Addresses Tab

Specifies a single IP address, range of IP addresses, or a subnet mask of users that can access a consumer. If IP addresses are specified for an consumer, only users who access the consumer from the appropriate IP addresses will be accepted by the consumer.

The IP Address tab of the Affiliate dialog contains the following fields:

### IP Address list

Lists IP address entries:

#### IP Address column

Lists IP addresses that can access the consumer. For IP address ranges, lists the first IP address in the range.

#### Subnet Mask column

Lists subnet masks of users that can access the consumer.

#### End of Range column

Lists the last IP addresses in ranges of IP addresses of users that can access the consumer. The beginning of the range of IP addresses is indicated by the value in the IP Address column of the same row in the table.

#### Name column

Lists host names of users that can access the consumer.

#### Add Button

Opens the Add an IP Address dialog from where you can add IP address entries.

### More Information:

[Configure IP Address Restrictions for 1.x Consumers \(optional\)](#) (see page 262)

## Affiliate Dialog--Time Tab

The Time tab is where you specify time restrictions for generating assertions for a consumer. When you add a time restriction, a user must attempts to access an consumer during the specified time period. If the user attempts to access a resource outside of the period, the producer does not generate a SAML assertion.

Configuring time restrictions for assertions is similar to configuring restrictions for policies.

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



The controls on the Time tab are as follows:

**Set button**

Opens the Time Dialog from where you can specify time restrictions for the consumer.

**Remove button**

Removes time restrictions for generating an assertion.

**More Information:**

[Configure Time Restrictions for 1.x Consumers \(optional\)](#) (see page 263)

## Affiliate Dialog--Advanced Tab

The Advanced tab is where you can optionally configure an Assertion Generator plug-in to customize the content of the SAML assertion generated by the Assertion Generator.

The fields on the Advanced tab of the Affiliate dialog are as follows:

**Full Java Class Name**

Defines the Java class name of the plug-in that will be invoked by the Assertion Generator at run time.

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 consumer. For example, `com.mycompany.assertiongenerator.AssertionSample`

**Parameters**

(Optional) Defines a string of parameters 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.

**More Information:**

[Customize SAML 1.x Assertion Content \(optional\)](#) (see page 263)



# Chapter 23: SAML 1.x Authentication Scheme Properties

---

This section contains the following topics:

[Authentication Scheme Dialog Prerequisites](#) (see page 523)

[Navigate to the Authentication Scheme Dialog](#) (see page 523)

[Authentication Scheme Dialog--SAML Artifact Template](#) (see page 524)

[Authentication Scheme Dialog--SAML Artifact Template--Scheme Setup Tab](#) (see page 525)

[Authentication Scheme Dialog--SAML Artifact Template--Advanced Tab](#) (see page 532)

[Authentication Scheme Dialog--SAML POST Template](#) (see page 532)

[Authentication Scheme Dialog--SAML POST Template--Scheme Setup Tab](#) (see page 533)

[Authentication Scheme Dialog--SAML POST Template--Advanced Tab](#) (see page 538)

[Authentication Scheme Dialog--SAML 1.x Auth Scheme Properties \(Artifact, POST\)](#) (see page 538)

## Authentication Scheme Dialog Prerequisites

To successfully configure a SAML 1.x authentication scheme (artifact or POST):

- You must have installed the Policy Server and Policy Server Option Pack.
- Your SiteMinder administrator account must have the Manage System and Domain Objects privilege.

## Navigate to the Authentication Scheme Dialog

You can access the Authentication Scheme dialog in the Policy Server User Interface.

**Note:** When you create or modify a Policy Server object in the Policy Server User Interface, use ASCII characters. Object creation or modification with non-ASCII characters is not supported.

To access the Authentication Scheme Dialog, do the following:

1. Begin at the menu bar.
2. Select Edit, System Configuration, Create Authentication Scheme

The Authentication Scheme Dialog opens.

To access the Authentication Scheme Dialog for an existing authentication scheme:

1. In the object pane, click on Authentication Schemes.
2. In the Authentication Scheme List pane, select an existing authentication scheme.
3. Right click and select Properties of Authentication Scheme.

## Authentication Scheme Dialog--SAML Artifact Template

If you have purchased SiteMinder Federation Security Services, you can configure the SAML artifact authentication scheme for SAML 1.x. After the scheme is configured, you can assign it to a realm.

### Scheme Common Setup Group Box

This group box lets you identify the authentication scheme.

#### Name

Designates a name for the authentication scheme.

**Note:** When Active Directory Application Mode (ADAM) is used as a policy store, the maximum length for the authentication scheme name is 22 characters.

#### Description

Provides a description of the authentication scheme.

#### Protection Level

Allows single sign-on for authentication schemes of equal or lower protection levels within the same policy domain, while requiring additional authentication to access resources with higher protection level schemes.

**Limits:** 1 and 1000.

Authentication schemes have a default protection level that you may need to change. High protection levels should be used for critical resources, while lower level schemes should be used for commonly accessible resources.

### More Information:

[Configure SiteMinder as a SAML 1.x Consumer](#) (see page 271)

[Authentication Scheme Dialog--SAML Artifact Template--Scheme Setup Tab](#) (see page 525)

[Authentication Scheme Dialog--SAML Artifact Template--Advanced Tab](#) (see page 532)

## Authentication Scheme Dialog--SAML Artifact Template--Scheme Setup Tab

The Scheme Setup tab for SAML 1.x artifact authentication is where you specify how the consumer communicates with the producer to retrieve the assertion, authenticate a user based on that assertion, then direct the user to the target resource.

The fields on the Scheme Setup tab of the SAML Artifact Template are as follows:

### Affiliate Name

Names of the consumer. Enter an alphabetic string, for example, CompanyA.

The name you enter here must match a name of a consumer in the Affiliate domain at the producer site.

**Important!** For the SAML artifact profile, the producer sends the assertion over a protected back channel to the consumer. If you are using basic authentication to protect the back channel, the value of this field is the name of the consumer. If you are using client certificate authentication for the back channel, the value of this field is the alias of the client certificate stored in the smkeydatabase.

### Redirect Mode drop-down list

Indicates the method by which the SAML credential collector redirects the user to the target resource. If you select 302 No Data or 302 Cookie Data, no other configuration is required. If you select Server Redirect or PersistAttributes, there is additional configuration required.

#### 302 No Data (default)

Redirects user through an HTTP 302 redirect with a session cookie, but no other data.

#### 302 Cookie Data

Redirects user through an HTTP 302 redirect with a session cookie and additional cookie data configured at the site that produced the assertion.

#### Server Redirect

Enables header and cookie attribute information, which is received as part of a SAML assertion, to be passed to the custom target application. The SAML Credential Collector transfers the user to the target application URL by using server-side redirect technology. Server-side redirects are part of the Java Servlet specification, and are supported by all the standard-compliant servlet containers.

To use this mode, you must follow these requirements:

- The URL you specify for this mode must be relative to the context of the servlet that is consuming the assertion, which is typically `/affwebservices/public/`. The root of the context is the root of the Federation Web Services application, typically `/affwebservices/`.

All target application files need to be in the application's root directory. This directory is either:

—Web Agent: `web_agent_home\webagent\affwebservices`

—SPS federation gateway:  
`sps_home\secure-proxy\Tomcat\webapps\affwebservices`

- You need to define realms, rules, and policies to protect target resources. The realms must be defined with at least the value `/affwebservices/` in the resource filter.
- You must have a custom Java or JSP application on the server that is serving the Federation Web Services application--that is, the server where the Web Agent Option Pack or SPS federation gateway is installed.

Java servlet technology allows applications to pass information between two resource requests using the `setAttribute` method of the `ServletRequest` interface.

The service that consumes assertions sends the user attribute to the target application by setting different attribute objects in the request object before redirecting the user to the target application.

The service creates two `java.util.HashMap` objects--one to store all header attributes and one to store all cookie attributes. It uses distinguished attribute names to represent each hashmap object:

—`Netegrity.HeaderAttributeInfo` attribute represents the hashmap that contains header attributes.

—`Netegrity.CookieAttributeInfo` attribute represents the hashmap that contains cookie attributes.

Two other `Java.lang.String` attributes are set by the assertion consuming service to pass the user identity to the custom application:

—`Netegrity.smSessionID` attribute represents the SiteMinder session ID

—`Netegrity.userDN` attribute represents the SiteMinder user DN.

The custom target application at the customer site can read these objects from the HTTP request object and can make use of the data found in the hashmap objects.

**PersistAttributes**

Redirects the user through an HTTP 302 redirect with a session cookie, but no other data. Additionally, this mode instructs the Policy Server to store attributes extracted from an assertion in the session store so they can be supplied as HTTP header variables. For additional configuration, see the instructions for using [SAML attributes as HTTP headers](#) (see page 288).

**Note:** If you choose PersistAttributes and the assertion contains attributes that are left blank, a value of NULL is written to the session store. This value acts as a placeholder for the empty attribute and it is passed to any application using the attribute.

**Password**

Defines the password that the consumer uses to identify itself to the producer.

This password must match the password entered for the consumer at the producer site.

**Verify Password**

Confirms the password that you specified in the Password field.

**Company Source ID**

Defines the source ID of the producer.

The SAML specification standard defines a source ID as a 20-byte binary, hex-encoded number that identifies the producer. The consumer uses this ID to identify an assertion issuer.

Enter a number that matches the Source ID that the producer uses. At the producer, this ID is located in the SAML assertion generator properties file, `AMAssertionGenerator.properties`, which is located in the following directory:

`policy_server_home/config/properties`

The default Company Source ID value is:

`b818452610a0ea431bff69dd346aeff83128b6a`

**SAML Version**

Specifies the version of the SAML specification with which the assertion is generated. The choices are 1.0 or 1.1. SAML 1.1 is the default.

There must be a match between the SAML protocol and assertion versions used by the producer or consumer. For example, if a producer using SAML 1.1 receives a SAML 1.0 request, it returns an error. If a consumer using SAML 1.1 receives a SAML 1.0 assertion embedded into a SAML 1.1 protocol element, it returns an error.

**Assertion Retrieval URL**

Defines the URL of the service at the producer where the consumer retrieves the SAML assertion. Specifically, this URL identifies the location of the Assertion Retrieval Service, which must be a URL accessed over an SSL connection.

If the assertion retrieval service at the producer is part of a realm using the Basic over SSL authentication scheme, the default URL is:

`https://idp_server:port/affwebservices/assertionretriever`

**idp\_server:port**

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

If the assertion retrieval service at the producer is part of a realm using the X.509 client certificate authentication scheme, the default URL is:

`https://idp_server:port/affwebservices/certassertionretriever`

**idp\_server:port**

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

**Authentication**

Specifies the authentication method for the realm at the producer that contains the Assertion Retrieval Service. The value of this field determines the type of credentials that the SAML credential collector at the consumer must provide to access the Assertion Retrieval Service and retrieve the SAML assertion.

The Assertion Retrieval Service retrieves the assertion from the Policy Server at the producer, and then sends it to the consumer across an SSL back channel. We recommend protecting the Assertion Retrieval Service.

Select one of the following:

- Basic Auth if the Assertion Retrieval Service at the producer is part of a realm protected by a Basic over SSL authentication scheme.
- Client Cert if the Assertion Retrieval Service at the producer is part of a realm protected by an X.509 client certificate authentication scheme.

When you select this option, configure access to the Assertion Retrieval Service with a client certificate.

- No Auth if the Assertion Retrieval Service is unprotected.



Depending on the authentication method you select, there can be some additional configuration required.

- When you select Basic Auth, no additional configuration is required at the producer or consumer, unless the certificate of the Certificate Authority that established the SSL connection is not in the consumer's smkeydatabase. If the appropriate certificate is not in the key database, you must import it.
- If you select Client Cert, there are several configuration steps at the consumer and the producer that you must complete to access the Assertion Retrieval Service with the client certificate.

For 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 FIPS-only installations use certificates only encrypted with FIPS 140-compatible algorithms.

### **Audience**

Defines the audience for the SAML assertion.

The Audience identifies the location of a document that describes the terms and conditions of the business agreement between the producer and the consumer. The administrator determines the audience at the producer site. It also must match the audience specified for the consumer at the producer.

The audience value must not exceed 1 K.

To specify the audience, enter a URL. This element is case sensitive. For example:

`http://www.companya.com/SampleAudience`

### **Issuer**

Identifies the producer that issues assertions for the consumer. This element is case sensitive.

For example, `producerA.ca.com`

The consumer accepts assertions from only this issuer. The administrator at the producer determines the issuer.

**Note:** The value that you enter for the issuer must match the value of the AssertionIssuerID at the producer site. This value is specified in the AMAssertionGenerator.properties file located in:  
`siteminder_home/Config/properties/AMAssertionGenerator.properties`

### Search Data XPATH

Defines an Xpath query that tells the authentication scheme where to look in the SAML assertion for data used to look up a user in the user directory. The Xpath query and the Search Specification value map data from a SAML assertion to a user store entry.

Xpath queries should not contain namespace prefixes. The following 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()
```

Select the namespace and click Edit to the right of the search specification scroll box. The SiteMinder Authentication Scheme Namespace Mapping dialog box opens. In this dialog, you map the data collected by the Xpath query to a namespace attribute.

Example:

The following query extracts the text of the Username attribute from the assertion:

```
"/Assertion/AttributeStatement/Attribute/AttributeValue/SMContent/SMlogin/Use  
rname/text()"
```

"//Username/text()" extracts the text of first Username element in the SAML assertion using abbreviated syntax.

Other examples:

```
"substring-after(/Assertion/AttributeStatement/Attribute/AttributeValue/  
SMContent/SMprofile/NVpair[1]/text(),"header:uid=")"
```

This query string extracts the text of the header attribute named "uid" configured as the first attribute in the Affiliate dialog at the producer site.

Using the following abbreviated syntax extracts the text of the header attribute named "uid" configured as the first attribute in the Affiliate dialog at the producer site using abbreviated syntax:

```
"substring-after(//SMprofile/NVpair[1]/text(),"header:uid=")"
```

### D-Sig Alias

Specifies the alias of the public key in the smkeydatabase that is used to verify the digital signature which signed the assertion. This alias is associated with the certificate in the smkeydatabase. When you list the certificates in the smkeydatabase, the type of certificate used for verification is a CertificateEntry or KeyEntry certificate.

### Namespace

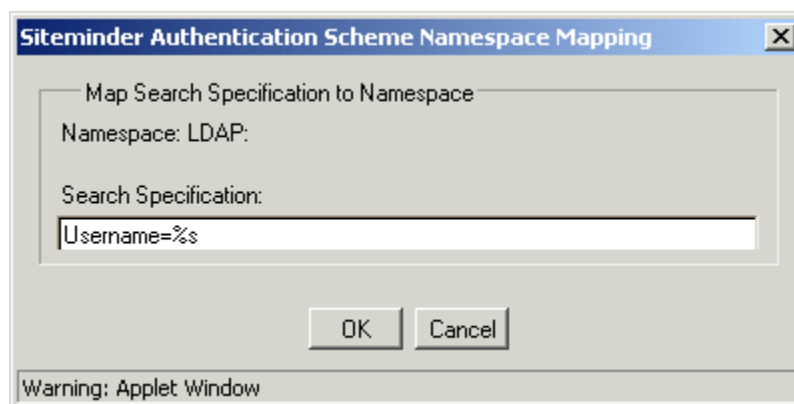
Selectable list of namespace types and defined search specifications.

**Search Specification**

Enter the attribute that the authentication scheme uses to search a namespace.

**Edit button**

Opens the Authentication Scheme Namespace Mapping dialog where you can enter a Search Specification which defines the attribute that the authentication scheme uses to search a namespace. Use %s in the entry as a variable representing the assertion data collected by the Search Data XPATH query.



For example, the XPATH query retrieves the value of user1 for the Username attribute in the SAML assertion. If you specify Username=%s in the Search Specification field, the resulting string is Username=user1. Compare this string against the user directory to find the correct record for authentication.

**Additional Configuration**

Selecting this button displays the SAML 1.x Auth Scheme Properties dialog for the authentication scheme. From this dialog, you define the message consumer plug-in, redirect URLs for failed authentication, and the target federation resource.

**More Information:**

[Authentication Scheme Dialog--SAML 1.x Auth Scheme Properties \(Artifact, POST\)](#) (see page 538)

[Configure SAML 1.x Artifact Authentication](#) (see page 280)

[Access the Assertion Retrieval Service with a Client Certificate \(optional\)](#) (see page 276)

[Configuration Settings that Must Use the Same Values](#) (see page 503)

## Authentication Scheme Dialog--SAML Artifact Template--Advanced Tab

The Advanced tab of the Authentication Scheme Dialog for an SAML artifact scheme is where you define information about the shared library that processes SAML 1.x artifact authentication.

The fields on the Advanced tab are as follows:

### Library

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 `smauthsaml`.

### Parameter

This field displays the information you entered in the Scheme Setup tab.

## Authentication Scheme Dialog--SAML POST Template

If you have purchased SiteMinder Federation Security Services, you can configure the SAML POST Template authentication scheme for the SAML 1.x POST profile. After configuring the authentication scheme and assign it to a realm.

### Scheme Common Setup Group Box

This group box lets you identify the authentication scheme.

#### Name

Designates a name for the authentication scheme.

**Note:** When Active Directory Application Mode (ADAM) is used as a policy store, the maximum length for the authentication scheme name is 22 characters.

#### Description

Provides a description of the authentication scheme.

#### Protection Level

Allows single sign-on for authentication schemes of equal or lower protection levels within the same policy domain, while requiring additional authentication to access resources with higher protection level schemes.

**Limits:** 1 and 1000.

Authentication schemes have a default protection level that you may need to change. High protection levels should be used for critical resources, while lower level schemes should be used for commonly accessible resources.

**More Information:**

[Configure SAML 1.x POST Profile Authentication](#) (see page 282)

[Authentication Scheme Dialog--SAML POST Template--Scheme Setup Tab](#) (see page 533)

[Authentication Scheme Dialog--SAML POST Template--Advanced Tab](#) (see page 538)

## Authentication Scheme Dialog--SAML POST Template--Scheme Setup Tab

The Scheme Setup tab for SAML 1.x POST is where you specify how the consumer communicates with the producer to retrieve the assertion, authenticate a user based on that assertion, then direct the user to the target resource.

The fields on the Scheme Setup tab of the SAML POST Template are as follows:

**Affiliate Name**

Names the consumer. Enter an alphabetic string, for example, CompanyA.

The name you enter here must match a name for a consumer in the Affiliate domain at the producer site.

**SAML Version**

Specifies the SAML version (inactive; the value defaults to 1.1, indicating that POST profile assertions are compliant with SAML version 1.1).

The SAML producer and consumer must be generating and consuming assertions and responses that are the same version.

**Audience**

Defines the audience for the SAML assertion.

The Audience identifies the location of a document that describes the terms and conditions of the business agreement between the producer and the consumer. The administrator determines the audience at the producer site. It also must match the audience specified for the consumer at the producer site.

The audience value must not exceed 1 K.

To specify the audience, enter a URL. This element is case sensitive. For example:

`http://www.ca.com/SampleAudience`

### Redirect Mode

Specifies the method by which the SAML credential collector servlet redirects the user to the target resource. If you select 302 No Data or 302 Cookie Data, no other configuration is required. If you select Server Redirect or PersistAttributes, there is additional configuration required.

#### 302 No Data

(Default). Redirects user through an HTTP 302 redirect with a session cookie, but no other data.

#### 302 Cookie Data

Redirects user through an HTTP 302 redirect with a session cookie and additional cookie data configured at the site that produced the assertion.

### Server Redirect

Enables header and cookie attribute information, which is received as part of a SAML assertion, to be passed to the custom target application. The SAML Credential Collector transfers the user to the target application URL by using server-side redirect technology. Server-side redirects are part of the Java Servlet specification, and are supported by all the standard-compliant servlet containers.

To use this mode, you must follow these requirements:

- The URL you specify for this mode must be relative to the context of the servlet that is consuming the assertion, which is typically `/affwebservices/public/`. The root of the context is the root of the Federation Web Services application, typically `/affwebservices/`.

All target application files need to be in the application's root directory. This directory is either:

—Web Agent: `web_agent_home\webagent\affwebservices`

—SPS federation gateway:

`sps_home\secure-proxy\Tomcat\webapps\affwebservices`

- You need to define realms, rules, and policies to protect target resources. The realms must be defined with at least the value `/affwebservices/` in the resource filter.
- You must have a custom Java or JSP application on the server that is serving the Federation Web Services application--that is, the server where the Web Agent Option Pack or SPS federation gateway is installed.

Java servlet technology allows applications to pass information between two resource requests using the `setAttribute` method of the `ServletRequest` interface.

The service that consumes assertions sends the user attribute to the target application by setting different attribute objects in the request object before redirecting the user to the target application.

The service creates two `java.util.HashMap` objects—one to store all header attributes and one to store all cookie attributes. It uses distinguished attribute names to represent each hashmap object:

- `Netegrity.HeaderAttributeInfo` attribute represents the hashmap that contains header attributes.

- `Netegrity.CookieAttributeInfo` attribute represents the hashmap that contains cookie attributes.

Two other `Java.lang.String` attributes are set by the assertion consuming service to pass the user identity to the custom application:

- `Netegrity.smSessionID` attribute represents the SiteMinder session ID

- `Netegrity.userDN` attribute represents the SiteMinder user DN.

The custom target application at the customer site can read these objects from the HTTP request object and can make use of the data found in the hashmap objects.

### **PersistAttributes**

User is redirected through an HTTP 302 redirect with a session cookie, but no other data. Additionally, this mode instructs the Policy Server to store attributes extracted from an assertion in the session store so they can be supplied as HTTP header variables. For additional configuration, see the instructions for using [SAML attributes as HTTP headers](#) (see page 288).

**Note:** If you choose `PersistAttributes` and the assertion contains attributes that are left blank, a value of `NULL` is written to the session store. This value acts as a placeholder for the empty attribute and it is passed to any application using the attribute.

### **Assertion Consumer URL**

Specifies the URL of the assertion consumer (synonymous with the SAML credential collector). This URL is where the user's browser must POST the generated assertion.

The default URL is:

`http://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. For example:

`http://www.discounts.com:85/affwebservices/public/samlcc`

### Issuer

Identifies the producer that issues assertions for the consumer. This element is case sensitive.

For example, producerA.ca.com

The consumer accepts assertions from only this issuer. The administrator determines the issuer at the producer.

**Note:** The value that you enter for the issuer must match the value of the AssertionIssuerID at the producer site. This value is specified in the AMAssertionGenerator.properties file located in:  
*policy\_server\_home/Config/properties/AMAssertionGenerator.properties*

### Dsig Issuer DN

Specifies the distinguished name of the certificate issuer that signs the SAML POST response. The SAML POST response must be digitally signed by the producer. When the consumer receives the response, it must verify the signature using the data in this parameter and the Serial Number parameter. These two parameters list the issuer of the certificate who signed the response.

The public key used to verify the signature must be in the smkeydatabase. More information about the smkeydatabase can be found in [Using Key Databases for Federation Security Services](#) (see page 479).

### Serial Number

Specifies the serial number (a hexadecimal string) of the certificate of the consumer in the smkeydatabase key store. This value is used with the Dsig Issuer DN to locate the certificate for digitally signing the SAML POST response.

### Search Data XPATH

Defines an XPATH query that the authentication scheme applies to the SAML assertion. The query tells the authentication scheme where to look in the assertion document. The data obtained by the query is used to look up a user in the user directory.

Xpath queries should not contain namespace prefixes. The following 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()
```



**Example**

The following query extracts the text of the Username attribute from the assertion:

```
"/Assertion/AttributeStatement/Attribute/AttributeValue/SMContent/SMlogin/Username/text()"
```

"//Username/text()" extracts the text of first Username element in the SAML assertion using abbreviated syntax.

Other examples:

```
"substring-after(/Assertion/AttributeStatement/Attribute/AttributeValue/SMContent/SMprofile/NVpair[1]/text(),"header:uid=")"
```

This query extracts the text of the header attribute named "uid" configured as the first attribute in the Affiliate dialog at the producer site. The string "substring-after(/SMprofile/NVpair[1]/text(),"header:uid=")" extracts the text of the header attribute named "uid" configured as the first attribute in the Affiliate dialog at the producer site using abbreviated syntax.

**Namespace**

Selectable List of namespace types and defined search specifications from which you can select namespace (user directory) type and then define a search specification for user disambiguation.

**Edit button**

Opens the Authentication Scheme Namespace Mapping dialog where you can enter a Search Specification, which maps data from a SAML assertion to a user store entry, which enables the SAML authentication scheme to locate the correct user record for authentication.

Use %s in the entry as a variable representing the assertion data collected by the Search Data XPATH query.

For example, the XPATH query retrieves the value of user1 for the Username attribute in the SAML assertion. If you specify Username=%s in the Search Specification field, the resulting string is Username=user1. This string is checked against the user directory to find the correct record for authentication.

**Additional Configuration**

Selecting this button displays the SAML 1.x Auth Scheme Properties dialog for the authentication scheme. From this dialog, you define the message consumer plug-in, redirect URLs for failed authentication, and the target federation resource.

**More Information:**

[Authentication Scheme Dialog--SAML 1.x Auth Scheme Properties \(Artifact, POST\)](#) (see page 538)

[Create the SAML 1.x POST Common Setup and Scheme Setup](#) (see page 282)

[Configuration Settings that Must Use the Same Values](#) (see page 503)

## Authentication Scheme Dialog--SAML POST Template--Advanced Tab

The Advanced tab of the Authentication Scheme Dialog for a SAML 1.x POST scheme is where you define information about the shared library that processes SAML 1.x POST authentication.

The fields on the Advanced tab are as follows:

### **Library**

This field contains the name of the shared library that processes SAML POST authentication. Do not change this value, unless you have a custom authentication scheme, written using the SiteMinder Authentication API.

The default shared library for SAML POST authentication is `smauthsaml`.

### **Parameter**

This field displays the information you entered in the Scheme Setup tab.

## Authentication Scheme Dialog--SAML 1.x Auth Scheme Properties (Artifact, POST)

The SAML 1.x Auth Scheme Properties dialog lets you configure additional features, such as the Message Extension Consumer API and optional redirect URLs for assertion processing errors during authentication. This dialog is also where you specify the target resource at the consumer site.

The dialog contains the following fields and controls:

### **Message Consumer Plugin Group Box**

#### **Full Java Class Name**

(Optional) Specifies the fully qualified Java class name of a class which implements a Message Consumer Plug-in interface for the authentication scheme.

#### **Parameter**

If a value is entered in the Full Java Class Name field, this field specifies a string of parameters the API passes to the specified plug-in.

### **Status Redirect URLs and Modes Group Box**

Assertion-based authentication can fail at the site that consumes assertions for various reasons. If authentication does fail, Federation Security Services provides functionality to redirect the user to different applications (URLs) for further processing. For example, when user disambiguation fails, SiteMinder can be configured to redirect the user to a provisioning system, which could create a user account based on the information found in SAML assertion.

The following options redirect the user to a configured URL based on the condition that caused the failure.

#### **Redirect URL for the User Not Found Status**

(Optional) Identifies the URL where SiteMinder redirects the user when the user is not found. The user not found status applies when the single sign-on message has no LoginID or the user directory does not contain the LoginID.

#### **Redirect URL for the Invalid SSO Message Status**

(Optional) Identifies the URL where SiteMinder redirects the user if one of the following conditions occur:

- The single sign-on message is not well formed, according to the XML specification
- The single sign-on message is invalid based on rules specified by the SAML schemas
- The consumer requires an encrypted assertion but the single sign-on message does not contain an encrypted assertion.

#### **Redirect URL for the Unaccepted User Credential (SSO Message) Status**

(Optional) Identifies the URL where SiteMinder redirects the user for all other error conditions other than when a user is not found or the single sign-on message is invalid. The assertion is valid, but SiteMinder does not accept the message for certain reasons, such as:

- XML digital signature validation fails
- XML decryption operation fails
- XML validation of conditions fails, such as an expired message or an audience mismatch
- None of the assertions in SSO message contain an authentication statement.

### **Mode**

Specifies the method by which SiteMinder redirects the user to the redirect URL. The options are:

#### **302 No Data (default)**

Redirects user with an HTTP 302 redirect with a session cookie, but no other data.

#### **Http Post**

Redirects user using HTTP Post protocol.

### **Target Page Configuration**

This section of the dialog lets you specify the URL of the target resource at the consumer site. If the query parameter is present, you can determine whether FSS replaces the value of the URL with the value of the TARGET query parameter in the authentication response URL.

#### **Default Target URL**

Specifies the URL of the target resource that resides at the consumer. This target is a protected federated resource that users can request.

#### **Query Parameter TARGET overrides Default Target URL**

(Optional) Replaces the value specified in the Default Target URL field with the value of the TARGET query parameter in the response. Using the TARGET query parameter, you can define the target dynamically and change the target with each authentication response. The flexibility of the TARGET query parameter offers more control over the target. In comparison, the Default Target URL value is a static value.

This check box is selected by default.

# Chapter 24: SAML 2.0 Service Provider Reference

---

This section contains the following topics:

[SAML Service Provider Dialog](#) (see page 541)  
[Navigate to the SAML Service Provider Dialog](#) (see page 541)  
[SAML Service Provider Dialog Prerequisites](#) (see page 542)  
[SAML Service Provider Properties Dialog Fields and Controls](#) (see page 542)  
[SAML Service Provider Dialog--Users Tab](#) (see page 545)  
[SAML Service Provider Dialog--Name IDs Tab](#) (see page 545)  
[SAML Service Provider Dialog--General Tab](#) (see page 547)  
[SAML Service Provider Dialog--SSO Tab](#) (see page 550)  
[Restrictions Dialog for an SP](#) (see page 553)  
[SAML Service Provider Dialog--Attributes Tab](#) (see page 555)  
[SAML Service Provider Dialog--SLO Tab](#) (see page 558)  
[SAML Service Provider Dialog--IPD Tab](#) (see page 559)  
[SAML Service Provider Dialog--Encryption Tab](#) (see page 560)  
[SAML Service Provider Dialog--Advanced Tab](#) (see page 561)  
[SAML 2.0--Additional URL Configuration Dialog](#) (see page 562)

## SAML Service Provider Dialog

If you have purchased SiteMinder Federation Security Services, you can use the SAML Service Provider Dialog to configure a SAML 2.0 Service Provider (SP).

## Navigate to the SAML Service Provider Dialog

Use one of the following methods to access the SAML Service Provider dialog in the Policy Server User Interface.

**Note:** When you create or modify a Policy Server object in the Policy Server User Interface, use ASCII characters. Object creation or modification with non-ASCII characters is not supported.

### To create a new SAML Service Provider

1. Begin at the menu bar.
2. Select Edit *affiliate\_domain\_name*, Create SAML Service Provider.

**To access an existing SAML Service Provider**

1. In the object pane, click on the Domains tab.
2. In the list pane, right-click the domain object you want to edit.
3. From the pop-up menu, select Properties of Domain.

## SAML Service Provider Dialog Prerequisites

The following prerequisites must be met in order to successfully create a policy or affiliate domain using the SAML Service Provider dialog:

- Your SiteMinder administrator account must have the Manage Domain Objects privilege.
- The Affiliate Domain in which the SAML Service Provider is to be configured must preexist.
- The Policy Server Session Server must be configured. The session store is where assertions are stored before they are forwarded to the Federation Web Services application.
- Federation Web Services must be installed on a system.

Federation Web Services can be installed by the Web Agent Option Pack or by the SPS federation gateway.

- The federated Service Provider site must be set up as a SAML 2.0 Service Provider. The SAML assertions generated at the Policy Server must be forwarded to an application at the Service Provider that can receive and interpret the assertions. The SAML Affiliate Agent and the SAML Credential Collector (installed with the Web Agent Option Pack or SPS federation gateway) can both act as SAML consumers.

**Note:** For information about the SAML Affiliate Agent, see the *SAML Affiliate Agent Guide*.

## SAML Service Provider Properties Dialog Fields and Controls

The SAML Service Provider Properties dialog contains the following fields and controls:

**Name**

Name of the Service Provider. This name should be unique across all affiliate domains.

**Description**

Optionally, a brief description of the Service Provider.

**Authentication URL**

Protected URL used to authenticate users trying to access Service Provider resources.

When a user who has not logged in at the Identity Provider requests a protected Service Provider resource, the user is sent to the Authentication URL. This URL must point to the redirect.jsp file, for example,

`http://myserver.idpA.com/siteminderagent/redirectjsp/redirect.jsp`

**myserver**

Identifies the web server with the Web Agent Option Pack or the SPS federation gateway. The redirect.jsp file is included with the Web Agent Option Pack or SPS federation gateway installed at the Identity Provider site.

**Important!** You must protect the AuthenticationURL; however, do this after you add a Service Provider to an affiliate domain because you have to exit the SAML Service Provider Properties dialog.

**Enabled**

Enables the Policy Server and Federation Web Services to support authentication of Service Provider resources.

**Use Secure URL**

The Use Secure URL feature instructs SiteMinder's Single Sign-on Service to encrypt only the SMPORTALURL query parameter that it appends to the Authentication URL prior to redirecting the user to establish a SiteMinder session. After the user is authenticated, the Authentication URL redirects the user back to the destination specified by the SMPORTALURL query parameter.

Encrypting the SMPORTALURL prevents a malicious user from modifying the value and redirecting authenticated users to a malicious web site instead of the originally requested target.

If you select this checkbox, you must do the following:

1. Set the Authentication URL field to the following URL:  
`http(s)://idp_server:port/affwebservices/secure/securedirect`
2. Protect the securedirect web service with a SiteMinder policy.

If the SiteMinder Producer/Identity Provider/Account Partner serves more than one Consumer/Service Provider/Resource Partner, it probably authenticates different users for these different partners. As a result, for each Authentication URL that uses securedirect, this web service must be included in a different realm for each partner.

To associate the securedirect service with different realms, modify the web.xml file to create different resource mappings; you cannot copy the securedirect web service to different locations on your server. This file is located in the directory `web_agent_home/affwebservices/WEB-INF`.

### **Application URL**

(Optional) Identifies the protected URL for a custom web application that is used to supply user attributes to the SiteMinder Single Sign-on service. The application can be on any host in your network.

Attributes from the web application specified in this field are made available to the Assertion Generator and then placed in the SAML assertion by an Assertion Generator plug-in. You must write and integrate the plug-in with SiteMinder.

The Federation Web Services application supplies sample web applications that you can use as a basis for your web application. They are:

`http://idp_server:port/affwebservices/public/sample_application.jsp`

`http://idp_server:port/affwebservices/public/unsolicited_application.jsp`

### **idp\_server:port**

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

The SAML Service Provider dialog also contains the following tabs:

### **Users tab**

Configures the users or groups of users for the Service Provider. Configured users can be authenticated for access to Service Provider resources; the assertion generator can create SAML assertions that include entitlement information for these users.

### **Name IDs tab**

Specifies the Name Identifier to be used when corresponding with this Service Provider.

### **General tab**

Specifies general information about the Service Provider.

### **SSO tab**

Configures the Single Sign-On (SSO) aspects of a Service Provider.

### **Attributes tab**

Configures assertion attribute statements, which define the user attributes, DN attributes, or static data that are passed from the Policy Server to the Service Provider in SAML 2.0 assertions.

### **SLO tab**

Configures the Single Logout (SLO) aspects of the Service Provider.

### **IPD tab**

Configures the Identity Provider Discovery Profile.



**Encryption tab**

Configures XML encryption.

**Advanced tab**

(Optional) Configures an Assertion Generator plug-in.

## SAML Service Provider Dialog--Users Tab

The Users tab is where you specify the users and groups that can be granted access to resources at the Service Provider. The Assertion Generator at the Identity Provider can generate assertions for these users for authentication purposes.

You can only add users and groups from directories that are included in the affiliate domain in which the Service Provider exists.

The controls on the Users tab are as follows:

**Add/Remove**

Opens the Users/Groups Dialog, which allows you to add and remove users and groups to the Users/Groups list box.

**Users/Groups**

Lists the users and groups that are included or excluded by the Policy that governs Assertion production for a particular user directory. The list box contains one tab for each user directory in the affiliate domain.

**More Information:**

[Select Users For Which Assertions Will Be Generated](#) (see page 303)

## SAML Service Provider Dialog--Name IDs Tab

The Name IDs tab is where you configure the Name Identifier, which names a user in a unique way in the assertion. The format of the Name Identifier establishes the type of content used for the ID. For example, the format might be the User DN, in which case the content might be a uid.

The Name IDs tab contains the following fields:

**Name ID Format**

Specifies the Name Identifier format. Pick one of the following:

- Unspecified
- EmailAddress

- X.509 Subject Name
- Windows Domain Qualified Name
- Entity Identifier
- Persistent Identifier
- Transient Identifier

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).

### **Name ID Type Group Box**

The Name ID group box contains radio buttons that specify the Name Identifier type. The choices are as follows:

#### **Static radio button**

Indicates that the Name Identifier is specified by the value in the Static Value field. Activates the Static Value field; disables other controls.

#### **User Attribute radio button**

Indicates that the Name Identifier resides in the user attribute specified in the Attribute Name field. Activates the Attribute Name field; disables other controls.

#### **DN Attribute radio button**

Indicates that the Name Identifier will be specified by an attribute associated with a DN. Activates the User Attribute field, the DN Spec field, and the Allow Nested Groups check box; disables the Static Value field.

#### **Allow Nested Groups check box**

Indicates that nested groups are allowed when selecting the DN. Enabled if the DN Attribute Radio Button is selected.

### **Name ID Fields Group Box**

Contains fields that specify information about the selected Name Identifier. The fields in this group box are context-sensitive, being determined according to the Name ID Type selection.

#### **Static Value field**

Specifies the static text value that will be used for all name identifiers for this Service Provider.

#### **User Attribute**

Specifies the name of the user attribute which contains the name identifier, or the attribute associated with a group or organizational unit DN.

**DN Spec**

Specifies the group or organizational unit DN to be used for obtaining the associated attribute to be used as the name identifier.

**Lookup button**

Opens the SiteMinder User Lookup dialog to locate the user group and select a DN to populate the DN Spec field.

**Other Controls****SAML Affiliation**

Specifies a SAML Affiliation for the Service Provider to join. Select from any configured SAML Affiliation object. If an Affiliation is selected, the other controls on the tab are grayed out (the Affiliation settings will be used instead).

**More Information**

[Specify Name Identifiers for SAML 2.0 Assertions](#) (see page 306)

## SAML Service Provider Dialog--General Tab

The General tab lets you specify general information about the Service Provider.

The settings on this tab are as follows:

**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. This URI value becomes the value of the Issuer field in the assertion.

### SAML Version

Specifies the SAML version (disabled; the value defaults to 2.0, indicating that assertions sent to this SP ID must be compliant with SAML version 2.0).

### Skew Time

Specifies the number of seconds (as a positive integer) added and subtracted from the current clock time to account for Service Providers with clocks that are not synchronized with the Policy Server acting as an Identity Provider. The skew time and the Validity Duration determine how SiteMinder FSS calculates the total time that an assertion is valid.

To determine the assertion validity, the skew time is subtracted from the assertion generation time (IssueInstant) to get the NotBefore time. The skew time is then added to the validity duration and the IssueInstant to get the NotOnOrAfter time. The following equations illustrate how the skew time is used:

- **NotBefore=IssueInstant - Skew Time**
- **NotOnOrAfter=Validity Duration + Skew Time + IssueInstant**

Times are relative to GMT.

For more information about assertion validity, see the *Federation Security Services Guide*.

### D-Sig Info Group Box

Enables you to specify digital signature processing information.

#### Disable Signature Processing

If set, all signature processing for this Service Provider (both signing and verification of signatures) is disabled for the Service Provider.

**Note:** Signature processing *must* be enabled in a production environment. Select the Disable Signature Processing option only for debugging purposes.

### Issuer DN

Specifies the distinguished name of the issuer of the Service Provider certificate used for signature verification of messages coming from that Service Provider. This value is used with the Serial Number to locate the certificate of the Service Provider in the SMKeyDatabase key store.

**Important!** This field is only enabled when either the HTTP Post option is set on the SSO tab or the HTTP Redirect Binding option is set on the SLO tab.

**Serial Number**

Specifies the serial number (a hexadecimal string) of the Service Provider certificate in the SmKeyDatabase key store. This certificate verifies the signature of messages coming from that Service Provider. This value is used with the Issuer DN to locate the certificate.

**Important!** This field is only enabled when either the HTTP Post option is set on the SSO tab or the HTTP Redirect Binding option is set on the SLO tab.

**Signing Options**

Displays the Signing Options dialog. From this dialog you can configure the settings for digital signing, such as the signing alias and the signature algorithm.

**Other Controls****Configure Backchannel Authentication**

Click this button to implement single sign-on with the HTTP-Artifact binding.

**More Information:**

[Configure Required General Information](#) (see page 308)

[Set a Password for SAML Artifact Back Channel Authentication](#) (see page 309)

[Validate Signed AuthnRequests and SLO Requests/Responses](#) (see page 309)

## SAML Service Provider Dialog--Backchannel Properties

The Backchannel Properties dialog is where you configure the password for the secure backchannel over which the Identity Provider sends the SAML assertion to the Service Provider.

The fields on this dialog are:

**Password**

Specifies the password to be used for Service Provider access across the back-channel.

**Confirm Password**

Confirmation of the entry in the Password field above.

**More Information:**

[Set a Password for SAML Artifact Back Channel Authentication](#) (see page 309)

## SAML Service Provider Dialog--SSO Tab

The SSO tab is where you configure Single Sign-On (SSO) information for the SAML Service Provider. The fields on this tab are as follows:

### Audience

Specifies the URI of the audience the IdP sends in the assertion (to be compared with the audience specified in the authentication scheme on the Service Provider). For example, sp.ca.com.

### Assertion Consumer Service

Specifies the URL of the service that receives assertions at the Service Provider. The default for SiteMinder is:

`http://<sp_server:port>/affwebservices/public/saml2assertionconsumer`

### sp\_server:port

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

**Note:** This field is disabled if indexed endpoints have been defined. In that case, you can only edit this URL by modifying an indexed entry in the Assertion Consumer Service dialog, accessed by selecting the ellipsis button.

### Ellipsis button

Opens the Assertion Consumer Services dialog. From this dialog, you can do the following:

- Add an indexed Assertion Consumer Service entry. You can add multiple entries; there is no limit.
- Edit an existing indexed Assertion Consumer Service entry
- Remove an existing indexed Assertion Consumer Service entry

**Bindings Group Box****HTTP-Artifact**

Indicates the artifact binding is supported at the Service Provider for single sign-on.

**Artifact Encoding**

Specifies the encoding used for the artifact binding: Form or URL. If you select URL, the artifact is added to a URL-encoded query string. If you select Form, the artifact is added to a hidden form control in a form.

**Override system generated IdP Source ID**

Allows you to specify an IdP Source ID in the associated field. The default is an SHA-1 hash of the IdP ID. Values must be a 40-digit hexadecimal number. To override this generated value, select this checkbox and enter a 40 character hexadecimal value for the IdP Source ID.

**Note:** If you select HTTP-Artifact, configure the backchannel settings on the General tab.

**HTTP-Post**

Indicates that the POST binding is supported at Service Provider for single sign-on.

**Other Controls****Enhanced Client and Proxy Profile**

Activates support for the SAML 2.0 Enhanced Client and Proxy (ECP) Profile. This profile is used if the SP and IdP are not communicating directly.

**Require Signed AuthnRequests**

Indicates that AuthnRequest messages sent by the Service Provider must be signed to be accepted. If you select this check box, the Identity Provider cannot send unsolicited responses, securing a trust between the Identity Provider and the Service Provider.

**Important!** If you enable this feature, you also have to complete the Issuer DN and Serial Number parameters on the General tab for validating the signature of the AuthnRequest.

**Signing Options**

Displays the Signing Options dialog. From this dialog you can configure the settings for digital signing, such as the signing alias and the signature algorithm.

**Authentication Level**

Specifies the minimum level at which the user must have authenticated to gain access to a SiteMinder realm. If the user has authenticated at this level or higher, the Identity Provider generates an assertion for the user. If the user is not authenticated at this level or higher, they are redirected to the Authentication URL to authenticate at this level.

### Validity Duration

Specifies a number of seconds (a positive integer) for which a generated assertion is valid.

In a test environment, you might want to 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)

**Note:** This property applies only to SSO messages -- it is *not* the same as setting specified in the Validity Duration field on the SLO tab.

### AuthnContext Class Ref

Defines the URI provided in the AuthnContextClassRef element to describe how the requesting user authenticated. Be sure that you specify a value that is legal based on the SAML specifications and is appropriate for the Authentication Level specified for the Service Provider. We recommend that you accept the default value, which is urn:oasis:names:tc:SAML:2.0:ac:classes:Password

You can specify another value, however, you then must configure a custom SAML response element on the Advanced tab.

### Allow Creation of New User Identifier

Permits the Identity Provider to create a value for the NameID in an assertion if the AuthnRequest message received from the Service Provider does not include a NameID but the AllowCreate attribute in the AuthnRequest message is set to true. The newly generated attribute for the NameID is included in the assertion.

### Restrictions button

Opens the Restrictions dialog from where you can configure IP address and time restrictions on the assertion generation policy.

### More Information:

[Configure Single Sign-on for SAML 2.0](#) (see page 311)

[Restrictions Dialog for an SP](#) (see page 553)



## Assertion Consumer Services Dialog

The Assertion Consumer Services dialog is where you configure an indexed Assertion Consumer Service entry.

The fields on this dialog are:

### Index

Specifies the index number for the URL.

**Limits:** unique integer from 0-65535.

### Binding

Specifies the binding you are using for single sign-on for this end point. Choose either HTTP-Artifact or HTTP-POST.

If single sign-on is initiated by an unsolicited request at the Identity Provider, and the link that triggers the request includes the ProtocolBinding query parameter, the binding specified this query parameter overrides the value you choose for this field.

### Assertion Consumer Service URL

Specifies the URL for the Assertion Consumer Service. The value must begin with URI must start with http:// or https: //

## Restrictions Dialog for an SP

The Restrictions dialog is where you configure IP address and time restrictions on the assertion generation policy for the Service Provider.

### More Information:

[Configure IP Address Restrictions for Service Providers \(optional\)](#) (see page 320)

[Configure Time Restrictions for Service Provider Availability \(optional\)](#) (see page 321)

## Navigate to the Restrictions Dialog

To access the Restrictions dialog in the Policy Server User Interface, click the Restrictions button on the SAML Service Provider dialog SSO tab.

**Note:** When you create or modify a Policy Server object in the Policy Server User Interface, use ASCII characters. Object creation or modification with non-ASCII characters is not supported.

### More Information:

[SAML Service Provider Dialog--SSO Tab](#) (see page 550)

## Restrictions Dialog Prerequisites

You must be in the process of configuring single sign-on for a Service Provider from the Service Provider Properties dialog.

## Restrictions Dialog Fields and Controls

The Restrictions dialog has the following fields and controls

### IP Addresses Group Box

#### IP Addresses

A list of restricted IP addresses configured for the policy for the Service Provider resources. The Policy Server User Interface 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.

#### Add

Opens an empty Add IP Address dialog from where you can create a new IP address restriction.

#### Edit

Opens the IP restriction entry selected in the IP Addresses list for editing in the Add IP Address dialog.

#### Remove

Deletes the IP restriction entry selected in the IP Addresses list.

### Time Restrictions Group Box

#### Set Button

Opens the Time dialog so you can configure the availability of the Resource Partner's availability. When you add a time restriction, the Service Provider functions only during the period specified.

#### More Information:

[Configure IP Address Restrictions for Service Providers \(optional\)](#) (see page 320)

[Configure Time Restrictions for Service Provider Availability \(optional\)](#) (see page 321)

## SAML Service Provider Dialog--Attributes Tab

The Attributes tab is where you configure SAML assertion attribute statements. You also indicate whether the attribute will be used for calls to the Attribute Authority or for single sign-on.

The fields on this dialog are as follows:

**Attribute**

Selectable list of configured attributes.

**Create**

Opens the Service Provider Attribute Editor dialog from where you can configure Service Provider entitlement attributes.

**Edit**

Opens the attribute entry selected in the Attribute List for editing in the SAML Service Provider Attribute Editor dialog.

**Remove**

Deletes the attribute entry selected in the Attribute List.

**Allow Nested Groups**

Allows nested groups to be searched in an LDAP Directory.

**More Information:**

[Configure Attributes for Inclusion in Assertions \(optional\)](#) (see page 329)

## Navigating to the SAML Service Provider Attribute Editor Dialog

To access the SAML Service Provider Attribute Editor dialog in the Policy Server User Interface, click the Create or Edit button on the SAML Service Provider dialog Attributes tab.

**Note:** When you create or modify a Policy Server object in the Policy Server User Interface, use ASCII characters. Object creation or modification with non-ASCII characters is not supported.

**More Information:**

[SAML Service Provider Dialog--Attributes Tab](#) (see page 555)

## SAML Service Provider Attribute Editor Dialog Prerequisites

You must be in the process of configuring a SAML Service Provider from the SAML Service Provider dialog.

## SAML Service Provider Attribute Editor Dialog--Attribute Setup Tab

### Attribute field

Specify the format for the attribute that will be part of a SAML assertion. Options are:

- unspecified
- basic
- uri

Refer to the SAML 2.0 specification for definitions of these formats.

### Attribute Kind Group Box

The Attribute Kind group box contains radio buttons that allow you to specify the attribute type:

#### Static

Returns data that remains constant. Use a static attribute to include as part of an assertion. Specify static values in the Variable Name and Variable Value fields.

#### User Attribute

Returns profile information from a user entry in a user directory. Specify the name for the attribute in the Variable Name field and the user directory attribute in the Attribute Name field.

**Note:** For the SiteMinder IdP to return values from a user directory, configure the user directory in the SiteMinder User Directory pane.

#### DN Attribute

Returns profile information from a directory object in a user directory. Groups 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.

Activates the Variable Name, DN Spec, and Attribute Name, which you configure to add the attribute to the assertion.

Selecting a DN Attribute also activates the Allow Nested Groups check box. Allowing nested groups lets SiteMinder 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 the Identity Provider to return an assertion containing DN attributes, the user directories must be configured in the SiteMinder User Directory pane.

**Allow Nested Groups**

Indicates that nested groups are allowed when selecting the DN. Enabled if the DN Attribute is selected.

**Note:** If the attribute is configured to be encrypted, the Encryption Certificate settings in the Encryption tab will be enabled.

**Encrypted**

Instructs SiteMinder to encrypt the DN attribute.

**Retrieval Method**

Specifies the intended use of the attribute.

**Limits:**

- SSO  
Indicates that the attribute is used for single sign-on.
- Attribute Service  
Indicates the attribute is for use by the Attribute Authority to complete requests from an attribute query.

**Attribute Fields**

Complete the fields for the Attribute Kind that you selected. Different fields become available depending on the attribute type (static, user attribute, DN attribute).

**Variable Name**

Designates the name for the attribute that SiteMinder returns in the assertion. Complete this field for any attribute type.

**Variable Value**

(Static only) Defines the static text as the value for the name/value pair. Enter a value.

**Attribute Name**

(User and DN Attributes only) Specifies the user directory attribute SiteMinder uses for the name/value pair.

**DN Spec**

(DN Attribute only) Specifies the DN of the user or user group and the name of the user attribute in the Attribute Name field.

## SAML Service Provider Attribute Editor Dialog--Advanced Tab

The Advanced tab of the Service Provider Attribute Editor dialog contains the following field:

### Script

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.

**Note:** If you copy and paste the contents of the Script field for another entitlement, you must select the appropriate radio button in the Attribute Kind group box of the Attribute Setup tab.

## SAML Service Provider Dialog--SLO Tab

The SLO tab is where you configure Single Logout (SLO) properties of the SAML Service Provider.

### HTTP-Redirect

Specifies whether the IdP-initiated Single Logout Profile over HTTP is supported at the Service Provider.

### Request Expiration Group Box

#### Validity Duration

Specifies the number of seconds for which a SLO request is valid.

**Note:** This property applies only to SLO messages -- it is *not* the same as that specified in the Validity Duration field on the SSO tab.

### URLs Group Box

#### SLO Location URL

(Required) Specifies the URL of the single logout service at the Service Provider. The default URL is:

`http://sp_server:port/affwebservices/public/saml2slo`

#### sp\_server:port

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

**SLO Response Location URL**

(Optional) Specifies the URL of the single logout service at the Service Provider. The purpose of having a Response Location URL is for a configuration where there is one service for single logout requests and one service for single logout responses.

For SiteMinder, this value is always the same as the SLO Location URL:

`http://sp_fws_server:port/affwebservices/public/saml2slo`

where *sp\_fws\_server:port* is the server at the Service Provider where the Federation Web Services application is installed.

For third-party vendors, the URL represents the service handling single logout responses.

**SLO Confirm URL**

Specifies the URL that the Identity Provider or Service Provider redirects the user when the single logout request is complete. This value needs to be a local resource and not a resource in a federated partner's domain. For example, if the local domain is acme.com, the SLO confirm page cannot be in the example.com domain.

**Relay State Overrides SLO Confirm URL**

(Optional) Replaces the URL in the SLO Confirm URL field with the value of the Relay State query parameter included with the single logout request to the SLO service. This check box gives you more control over the single logout confirmation target because using the Relay State query parameter lets you dynamically define the confirmation URL for SLO requests.

## SAML Service Provider Dialog--IPD Tab

The IPD tab is where you configure the Identity Provider Discovery profile (IPD), which enables the Service Provider to determine which Identity Provider the principals are using with the Web browser single sign-on profile.

The fields on this dialog are as follows:

**Enable**

Indicates that the Identity Provider Discovery profile is enabled.

**Service URL**

Specifies the URL of the Identity Provider Discovery Profile servlet. The default URL is:

`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.

**Common Domain**

Specifies the common cookie domain (must be a subset of the host in the Service URL).

**Persistent Cookie**

Indicates that the cookie should be persistent.

## SAML Service Provider Dialog--Encryption Tab

The Encryption tab is where you configure XML encryption for a SAML assertion. If you enable encryption, all data in the assertion is encrypted, including all attribute statements.

**Note:** If you do not want to encrypt the entire assertion but want to encrypt only individual attribute statements, go to the Attributes tab, select or create an attribute, and check the Encrypted check box for the individual attribute.

The fields on the encryption tab are as follows:

**Encrypt Name ID**

Specifies the Name ID in the assertion is encrypted.

**Encrypt Assertion**

Enables encryption of the assertion.

**Encryption Block Algorithm**

Specifies the block algorithm to be used for encryption. Choose one of the following:

- triledes (the default)
- aes-128
- aes-256

**Encryption Key Algorithm**

Specifies the key algorithm to be used for encryption. Choose one:

- rsa-v15 (the default)
- rsa-oaep

**Note:** The minimum key size required to use the rsa-oaep encryption algorithm is a 1024 bits.



**Encryption Public Key Certificate Group Box**

This group box is where you specify the location of the public certificate of the Service Provider.

**Note:** You must complete both fields in this group box if either the Encrypt Name ID or Encrypt Assertion option is set or any SAML assertion attribute needs encryption.

**Issuer DN**

Specifies (along with the Serial Number) the Issuer DN of the certificate of the Service Provider in the key store.

**Serial Number**

Specifies (along with the Issuer DN) the serial number of the certificate of the Service Provider in the key store.

**More Information:**

[Encrypt a NameID and an Assertion](#) (see page 336)

## SAML Service Provider Dialog--Advanced Tab

The Advanced tab is where you optionally configure a custom-developed Assertion Generator plugin developed using the Assertion Generator Plugin API.

**Assertion Generator Plugin Group Box****Full Java Class Name**

Specifies the fully qualified Java class name of the Assertion Generator Plugin.

**Parameters**

If a value is entered in the Full Java Class Name field, specifies a string of parameters to be passed to the specified plugin.

### Proxy Group Box

#### Server

If your network has a proxy server between the client and the system where Federation Web Services is running (that is, the system with the Web Agent Option Pack or the SPS federation gateway) specify the scheme and authority portions of the URL, such as *protocol:authority*. The scheme is http: or https: and the authority is *//host.domain.com* or *//host.domain.com:port*. For example, *http://example.ca.com*.

### Status Redirect URLs and Modes Group box

#### Additional URL Configuration

Allows you to specify additional redirect URLs for HTTP status errors. The user can be redirected to a custom error page for further action.

Click this button to open the Additional URL Configuration dialog, which is described in the next topic.

## SAML 2.0--Additional URL Configuration Dialog

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 Service Provider is configured at the Identity Provider. For example, if during a request there is an issue in retrieving the certificate information from smkeydatabase, the user is redirected to Server Error URL specified. However, if a request contains an invalid SP ID, no redirection happens and the HTTP error code 400 is returned to the browser.

The Additional URL Configuration dialog lets you configure redirect URLs for error handling. This dialog includes the following settings:

#### Enable Server Error URL

Specifies the URL where the user is redirected when an HTTP 500 Server error occurs. A user can encounter a 500 error because an unexpected condition prevents the web server from fulfilling the client request. If this type of error occurs, the user is sent to the specified URL for further processing.

Example: *http://www.redirectmachine.com/error\_pages/server\_error.html*

**Enable Invalid Request URL**

Specifies the URL where the user is redirected when an HTTP 400 Bad Request or a 405 Method Not Allowed error occurs. A user can encounter a 400 error because a request is malformed or a 405 error because the web server does not allow a particular method or action to be performed. If these types of errors occur, the user is sent to the specified URL for further processing.

Example: [http://www.redirectmachine.com/error\\_pages/invalidreq\\_error.html](http://www.redirectmachine.com/error_pages/invalidreq_error.html)

**Enable Unauthorized Access URL**

Specifies the URL where the user is redirected when an HTTP 403 Forbidden error occurs. A user can encounter a 403 error because the URL in a request is pointing to the wrong target, such as a directory instead of a file. If this error occurs, the user is sent to the specified URL for further processing.

Example:

[http://www.redirectmachine.com/error\\_pages/unauthorized\\_error.html](http://www.redirectmachine.com/error_pages/unauthorized_error.html)

**Mode Fields**

For each URL, you can select a mode by which the browser redirects the user.

**302 No Data**

Redirects the user using an HTTP 302 redirect. When the user is redirected to the specified custom error page URL, they are redirected only with a session cookie and no information appended to the redirect URL.

**HTTP POST**

Redirects the user using the HTTP POST protocol. When the user is redirected to the custom error page URL, the following data, if available, is appended to the redirect URL and then posted to custom error page:

- TARGET
- AUTHREASON
- CONSUMERURL
- SAMLRESPONSE
- IDPID
- SPID
- PROTOCOLBINDING



# Chapter 25: SAML 2.0 Authentication Scheme Properties Reference

---

This section contains the following topics:

[Authentication Scheme Dialog -- SAML 2.0 Template](#) (see page 565)  
[Navigate to the SAML 2.0 Auth Scheme Properties Dialog](#) (see page 566)  
[SAML 2.0 Authentication Properties--Scheme Setup Tab](#) (see page 566)  
[SAML 2.0 Authentication Properties--Advanced Tab](#) (see page 568)  
[SAML 2.0 Auth Scheme Properties Dialog](#) (see page 568)  
[SAML 2.0 Auth Scheme Properties Dialog Prerequisites](#) (see page 568)  
[SAML 2.0 Auth Scheme Properties Dialog Fields and Controls](#) (see page 568)  
[SAML 2.0 Auth Scheme Properties Dialog--Users Tab](#) (see page 569)  
[SAML 2.0 Auth Scheme Properties Dialog--SSO Tab](#) (see page 571)  
[SAML 2.0 Auth Scheme Properties Dialog--SLO Tab](#) (see page 575)  
[SAML 2.0 Auth Scheme Properties Dialog--Encryption Tab](#) (see page 577)  
[SAML 2.0 Auth Scheme Properties Dialog--Attributes Tab](#) (see page 577)  
[SAML 2.0 Auth Scheme Properties Dialog--Name IDs Tab](#) (see page 577)  
[SAML 2.0 Auth Scheme Properties Dialog--Backchannel Tab](#) (see page 578)  
[SAML 2.0 Auth. Scheme Properties Dialog--Advanced Tab](#) (see page 579)  
[SAML 2.0--Additional URL Configuration](#) (see page 581)

## Authentication Scheme Dialog -- SAML 2.0 Template

If you have purchased SiteMinder Federation Security Services, you can configure the SAML 2.0 Template authentication scheme.

### Scheme Common Setup Group Box

This group box lets you identify the authentication scheme.

#### Name

Designates a name for the authentication scheme.

**Note:** When Active Directory Application Mode (ADAM) is used as a policy store, the maximum length for the authentication scheme name is 22 characters.

#### Description

Provides a description of the authentication scheme.

#### Protection Level

Allows single sign-on for authentication schemes of equal or lower protection levels within the same policy domain, while requiring additional authentication to access resources with higher protection level schemes.

**Limits:** 1 and 1000.

Authentication schemes have a default protection level that you may need to change. High protection levels should be used for critical resources, while lower level schemes should be used for commonly accessible resources.

## Navigate to the SAML 2.0 Auth Scheme Properties Dialog

To access the SAML 2.0 Auth Scheme Properties dialog in the Policy Server User Interface, click the Additional Configuration button on the Authentication Scheme Dialog--SAML 2.0 Template--Scheme Setup Tab.

**Note:** When you create or modify a Policy Server object in the Policy Server User Interface, use ASCII characters. Object creation or modification with non-ASCII characters is not supported.

## SAML 2.0 Authentication Properties--Scheme Setup Tab

The Scheme Setup tab for SAML 2.0 authentication lets you specify how the Service Provider communicates with the Identity Provider to retrieve the assertion.

The fields on this tab are:

#### SP ID

Specifies a URI that uniquely identifies the Service Provider.

**Note:** The value that you enter must match the value of the ID specified for the corresponding Service Provider object that you configure at the Identity Provider.

#### IdP ID

Specifies a URI that uniquely identifies the Identity Provider from which assertions for this Service Provider are issued.

The Service Provider accepts assertions from only this IdP.

**Note:** The value that you enter for the issuer must match the value of the IdP ID configured at the Identity Provider site.

#### SAML Version

Specifies the SAML version (disabled; the value defaults to 2.0, indicating that assertions sent to this IdP ID must be compliant with SAML version 2.0).

**D-Sig Info Group Box**

Contains fields and controls that allow you to specify digital signature information:

**Disable Signature Processing**

Disables all signature processing, that is, signing and verification of signatures, for this Service Provider.

**Note:** Signature processing *must* be enabled in a production environment. Select the Disable Signature Processing option only for debugging purposes.

**Signing Options**

Displays a dialog with the settings for digital signature configuration, specifically the Signing Alias and the Signature Algorithm.

**Issuer DN**

Specifies the distinguished name of the issuer of the certificate. This value is used with the Serial Number to locate the certificate of the Service Provider in the SMKeyDatabase file.

**Note:** This field is enabled only if the HTTP Post option is set on the SSO tab or the HTTP Redirect Binding option is set on the SLO tab. If signature processing is disabled, this field is inactive.

**Serial Number**

Specifies the serial number (a hexadecimal string) of the certificate of the Service Provider in the key store. This value is used with the IssuerDN to locate the certificate in the SMKeyDatabase key store.

**Note:** This field is enabled only if the HTTP Post option is set on the SSO tab or the HTTP Redirect Binding option is set on the SLO tab. If signature processing is disabled, this field is inactive.

**Additional Configuration**

Opens the SAML 2.0 Auth Scheme Properties dialog. From this dialog, you can specify additional configuration details for the authentication scheme.

## SAML 2.0 Authentication Properties--Advanced Tab

The Advanced tab of the Authentication Scheme Properties dialog for SAML 2.0 is where you define information about the shared library that processes SAML 2.0 authentication.

### **Library**

Contains the name of the shared library that processes SAML 2.0 authentication. Do not change this value, unless you have a custom authentication scheme, written using the SiteMinder Authentication API.

The default shared library for SAML 2.0 authentication is `smauthsaml`.

### **Parameter**

Displays the information you entered in the Scheme Setup tab.

## SAML 2.0 Auth Scheme Properties Dialog

The SiteMinder SAML 2.0 Auth Scheme Properties dialog is where you specify configuration details for the SAML 2.0 Template authentication scheme.

### **More Information:**

[Configure SiteMinder as a SAML 2.0 Service Provider](#) (see page 353)

## SAML 2.0 Auth Scheme Properties Dialog Prerequisites

You must be in the process of configuring a SAML 2.0 Template Authentication scheme from the Authentication Scheme dialog.

## SAML 2.0 Auth Scheme Properties Dialog Fields and Controls

The SAML 2.0 Auth. Scheme Properties dialog contains the following tabs:

### **User**

Defines how to collect user information from information in a SAML 2.0 assertion.

### **SSO**

Specifies the Single Sign-On (SSO) bindings supported by the Service Provider.

### **SLO**

Determines the Single Logout (SLO) properties for the SAML Service Provider.



**Encryption**

Defines the XML encryption configuration for SAML assertions sent to this Service Provider.

**Attributes**

Specifies the attributes that the SAML Requester can request from an Attribute Authority.

**NameIDs**

Specifies the Name ID that will be in an attribute query sent from the SAML requester (SP) to the Attribute Authority (IdP).

**Backchannel**

Defines the configuration of the secure back channel between the Service Provider and the Identity Provider.

**Advanced**

Specifies advanced configuration for the Message Extension Consumer API and optional redirect URLs for assertion processing errors.

## SAML 2.0 Auth Scheme Properties Dialog--Users Tab

The Users tab is where you configure how to obtain the user information from a SAML 2.0 assertion and use this information to authenticate a user.

**User Disambiguation Group Box****Xpath Query**

Specifies an XPath query that the authentication scheme applies to the assertion to obtain the LoginID.

The default XPath query used when none is configured, is:

```
/Assertion/Subject/NameID/text()
```

Xpath queries should not contain namespace prefixes. The following 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()
```

### Example

To obtain the attribute called "FirstName" from the assertion for authentication, the XPath query is:

```
/Assertion/AttributeStatement/Attribute[@Name="FirstName"]/  
AttributeValue/text()
```

### Namespace

Displays a selectable list of namespace types and defined search specifications from which you can select namespace (user directory) type and then define a search specification for user disambiguation.

### Edit

Opens the Authentication Scheme Namespace Mapping dialog where you can enter a Search Specification which defines the attribute that the authentication scheme uses to search a namespace. Use %s as the entry representing the LoginID.

For example, the LoginID is 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.

### SAML Affiliation

(Optional) specifies a SAML Affiliation for the Identity Provider to join. Select from any configured SAML Affiliation object. If an Affiliation is selected, the remaining controls are dimmed and the Affiliation settings are used instead.

### User Disambiguation Group Box

The Authentication Scheme Namespace Mapping dialog box is where you specify the attribute that the authentication scheme uses to search a namespace.

### Search Specification

Specifies the attribute that the SAML 2.0 authentication scheme uses to search a namespace. Use %s in the entry as a variable representing the LoginID.

**More Information:**

[Access the Artifact Resolution Service with a Client Certificate \(optional\)](#) (see page 390)

## SAML 2.0 Auth Scheme Properties Dialog--SSO Tab

The SSO tab is where you configure Single Sign-On (SSO) information.

**Redirect Mode**

Specifies the method by which the Service Provider redirects the user to the target resource. If you select 302 No Data or 302 Cookie Data, no other configuration is required. If you select Server Redirect or PersistAttributes, there is additional configuration required.

**302 No Data (default)**

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

**302 Cookie Data**

User is redirected via an HTTP 302 redirect with a session cookie and additional cookie data configured for the Service Provider at the Identity Provider.

**Server Redirect**

Enables header and cookie attribute information, which is received as part of a SAML assertion, to be passed to the custom target application. The service that collects the credentials (SAML 2.0 Assertion Consumer Service or WS-Federation Security Token Consumer Service) transfers the user to the target application URL by using server-side redirect technology. Server-side redirects are part of the Java Servlet specification, and are supported by all the standard-compliant servlet containers.

To use this mode, you must follow these requirements:

- The URL you specify for this mode must be relative to the context of the servlet that is consuming the assertion, which is typically `/affwebservices/public/`. The root of the context is the root of the Federation Web Services application, typically `/affwebservices/`.

All target application files need to be in the application's root directory. This directory is either:

—Web Agent: `web_agent_home\webagent\affwebservices`

—SPS federation gateway:  
`sps_home\secure-proxy\Tomcat\webapps\affwebservices`

- You need to define realms, rules, and policies to protect target resources. The realms must be defined with at least the value `/affwebservices/` in the resource filter.

- You must have a custom Java or JSP application on the server that is serving the Federation Web Services application--that is, the server where the Web Agent Option Pack or SPS federation gateway is installed.

Java servlet technology allows applications to pass information between two resource requests using the `setAttribute` method of the `ServletRequest` interface.

The service that consumes assertions sends the user attribute to the target application by an attribute object in the request before redirecting the user to the target application. The service that consumes assertions sends the attributes by creating a `java.util.HashMap` object. The attribute that contains the `HashMap` of SAML attributes is "Netegrity.AttributeInfo."

Two other `Java.lang.String` attributes are set by the service that consumes assertions to pass the user identity to the custom application:

—`Netegrity.smSessionID` attribute represents the SiteMinder session ID

—`Netegrity.userDN` attribute represents the SiteMinder user DN.

The custom target application at the customer site can read these objects from the HTTP request object and can make use of the data found in the `hashmap` objects.

#### **PersistAttributes**

User is redirected via an HTTP 302 redirect with a session cookie, but no other data. Additionally, this mode instructs the Policy Server to store attributes extracted from an assertion in the session store so they can be supplied as HTTP header variables. For additional configuration, see the instructions for using [SAML attributes as HTTP headers](#) (see page 288).

**Note:** If you choose `PersistAttributes` and the assertion contains attributes that are left blank, a value of `NULL` is written to the session store. This value acts as a placeholder for the empty attribute and it is passed to any application using the attribute.

#### **SSO Service**

Specifies the URI of the Single Sign-On service at an Identity Provider. This is the location where the `AuthnRequest` service redirects an `authnrequest` message, which contains the Service Provider's ID. The default URL is:

`http://idp_host:port/affwebservices/public/saml2sso`

**Audience**

Specifies the audience for the SAML assertion. The Audience is a URL that identifies the location of a document that describes the terms and conditions of the business agreement between the Identity Provider and the Service Provider. The audience is determined by the administrator at the Identity Provider site. It also must match the audience specified for the Service Provider at the Identity Provider site.

The audience value should not exceed 1K and is case-sensitive. For example:

`http://www.ca.com/SampleAudience`

**Target**

Specifies the target resource URI at the destination Service Provider site.

**Relay State Overrides Target**

(Optional) Replaces the value specified in the Target field with the value of the Relay State query parameter for SP-initiated or IdP-initiated single sign-on. This check box gives you more control over the target because using the Relay State query parameter lets you dynamically define the target.

**Bindings Group Box****HTTP-Artifact**

Enables the artifact binding (when enabled, the following associated controls are activated).

**Override system generated IdP Source ID**

Allows you to specify an IdP Source ID in the associated field. The default is an SHA-1 hash of the IdP ID. Values must be a 40-digit hexadecimal number.

**Resolution Service**

Specifies the URL of the Identity Provider's Artifact Resolution Service. The default URL is:

`http://host:port/affwebservices/saml2artifactresolution`

**Sign ArtifactResolve**

Indicates that the artifact resolve message, the request sent by the Service Provider to retrieve the original SAML message, must be signed. If you check this box, the artifact resolve message must be signed or the Identity Provider will not accept it.

If you select this check box, the Identity Provider must be configured to require a signed artifact resolve message.

**Note:** Digital signature processing must be enabled to sign the artifact resolve message.

### **Require Signed ArtifactResponse**

Specifies that the Service Provider will only accept the artifact response, which contains the original SAML message, if the response is signed.

If you select this check box, the Identity Provider must be configured to sign the artifact response.

**Note:** Digital signature processing must be enabled to process the signed response.

### **Authentication**

Specifies the type of authentication that protects the realm that contains the Artifact Resolution Service. The authentication scheme determines the type of credentials the Service Provider must present to access the Artifact Resolution Service to retrieve the assertion.

Choose one:

#### **Basic**

(Default) Uses the SP ID and the password specified in the SSO tab of the authentication scheme properties for credentials. No additional configuration is required, unless the connection to the Artifact Resolution Service is an SSL connection. Then, the certificate of the Certificate Authority who enabled the SSL connection must be in the Service Provider's AM.keystore.

#### **Client Cert**

Uses the SP ID and password specified in the SSO tab of the authentication scheme properties for credentials to look up the certificate in the key store. If you select this option, you have to configure access to the Artifact Resolution Service using a client certificate.

#### **No Auth**

Indicates that no authentication is required.

### **Index**

Enabled upon selecting HTTP-Artifact check box, this field assigns an AssertionConsumerServiceIndex parameter for the artifact binding. If you have multiple endpoints in a federated network, you can assign an index for the Assertion Consumer Service so the Identity Provider knows where to send the response. Enter an integer in the range of 0-65535.

**Http-Post**

Indicates that the POST binding is enabled for the Identity Provider.

**Enforce Single Use Policy check box**

Enforces the single use policy is enforced, preventing SAML 2.0 assertions that arrive via POST binding from being re-used at a Service Provider to establish a second session.

**Index**

Enabled upon selecting HTTP-Post check box, this field assigns an AssertionConsumerServiceIndex parameter for the artifact binding. If you have multiple endpoints in a federated network, you can assign an index for the Assertion Consumer Service so the Identity Provider knows where to send the response. Enter an integer in the range of 0-65535.

**Other Controls****Enhanced Client and Proxy Profile**

Enables processing of requests using the SAML 2.0 Enhanced Client and Proxy (ECP) Profile.

**Sign AuthRequests**

Enables the signature of AuthnRequest messages (which request authentication from an Identity Provider).

**Allow IdP to Create New User Identifier**

If the Service Provider sends an AuthnRequest message to the Identity Provider to get an assertion, checking this box sets the AllowCreate attribute in the AuthnRequest message to true. The AllowCreate attribute instructs the Identity Provider to generate a new value for the NameID, provided that the AllowCreate feature is enabled at the Identity Provider. This new value for the NameID is included in the assertion sent back to the Service Provider.

## SAML 2.0 Auth Scheme Properties Dialog--SLO Tab

The SLO tab is where you configure Single Logout (SLO) properties for the SAML Service Provider being protected by the authentication scheme.

**Bindings Group Box****HTTP-Redirect**

Specifies whether the IdP-initiated Single Logout Profile over HTTP is supported at the Service Provider.

### **Request Expiration Group Box**

#### **Validity Duration**

Specifies the number of seconds for which a SLO request is valid.

**Note:** This property is different from that specified in the Validity Duration field on the SSO tab).

#### **Other Controls**

#### **SLO Location URL**

Required. Specifies the URL of the single logout service at the Identity Provider. The default URL is:

`http://idp_server:port/affwebservices/public/saml2slo`

#### **idp\_server:port**

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

#### **SLO Response Location URL**

Optional. Specifies the URL of the single logout service at the Identity Provider. The purpose of having a Response Location URL is for a configuration where there is one service for single logout requests and one service for single logout responses.

For SiteMinder, this value is always the same as the SLO Location URL:

`http://idp_fws_server:port/affwebservices/public/saml2slo`

#### **idp\_server:port**

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

For third-party vendors, the URL will represent the service handling single logout responses.

#### **SLO Confirm URL**

Specifies the URL that the Identity Provider or Service Provider redirects the user when the single logout request is complete. This value needs to be a local resource and cannot be a resource in a federated partner's domain. For example, if the local domain is ca.com, the SLO confirm page cannot be in the example.com domain.

#### **Relay State Overrides SLO Confirm URL**

(Optional) Replaces the URL in the SLO Confirm URL field with the value of the Relay State query parameter included with the single logout request to the SLO service. This check box gives you more control over the single logout confirmation target because using the Relay State query parameter lets you dynamically define the confirmation URL for SLO requests.



## SAML 2.0 Auth Scheme Properties Dialog--Encryption Tab

The Encryption tab is where you configure XML Encryption configuration for SAML assertions sent to this Service Provider.

Encryption Restrictions Group Box

### **Require Encrypted Name ID**

Indicates that the Name ID in the assertion must be encrypted. If the Name ID is not encrypted, the assertion will not be accepted.

### **Require Encrypted Assertion**

Indicates that the entire assertion must be encrypted. If the assertion is not encrypted, it will not be accepted.

Decryption Private Key Group Box

### **Alias**

Specifies an alias for the private key in the smkeydatabase used to decrypt encrypted data in the assertion received from the Identity Provider.

## SAML 2.0 Auth Scheme Properties Dialog--Attributes Tab

In the Attributes tab, you configure the attributes that the SAML requester can request from an Attribute Authority. These are the attributes that the SAML requester can include in the attribute query and are then returned by the Attribute Authority in its response.

A description of the contents of this tab can be found in the [SAML 2.0 Attribute Query Reference](#) (see page 591).

## SAML 2.0 Auth Scheme Properties Dialog--Name IDs Tab

The Name IDs tab is where you configure the Name ID to be included in the attribute queries sent from the SAML requester (SP) to the Attribute Authority (IdP).

This tab is only configurable if the Attribute Query feature is enabled, that is, the Enabled checkbox on the Attributes tab is checked.

A description of the contents of this tab can be found in Name IDs Dialog Fields and Controls for Attribute Queries (see page 594).

### **More Information:**

[Name IDs Dialog Fields for Attribute Queries](#) (see page 594)

## SAML 2.0 Auth Scheme Properties Dialog--Backchannel Tab

The Backchannel tab defines the configuration of the secure back channel for two functions:

- Single Sign-on between a Service Provider and Identity Provider
- Attribute Queries and Responses between a SAML Requester and an Attribute Authority

The following required fields serve the same function for both purposes, as follows:

### Authentication

Specifies the authentication method used across the back channel. Options are:

#### Basic (over SSL)

Indicates that the Single Sign-on or Attribute Service is part of a realm protected by a Basic over SSL authentication scheme.

If you select this option, no additional configuration other than the remaining required fields in this dialog, is required unless you want to use Basic over SSL. If you select Basic over SSL, you must ensure that the certificate of the Certificate Authority that enabled the SSL connection is in the smkeydatabase. If it is not, import the certificate into the smkeydatabase.

#### Client Cert

Indicates that the Single Sign-on or Attribute Service is part of a realm protected by an X.509 client certificate authentication scheme.

For single sign-on, if you select this option, you must also configure access to the Assertion Retrieval Service with a client certificate.

You can use non-FIPS 140 encrypted certificates to secure the backchannel even if the Policy Server is operating in FIPS-only mode. However, for FIPS-only installations use certificates only encrypted with FIPS 140-compatible algorithms.

#### NoAuth

Indicates that the Single Sign-on or Attribute Service is not protected. If you select this option, no authentication is required.

### SP Name

Identifies the Service Provider object. This name must match a Service Provider or SAML Requester name specified at the Identity Provider or Attribute Authority.

If you are using basic authentication as the authentication scheme for the backchannel, the value of this field is the name of the Service Provider. If you are using client certificate authentication for the backchannel, the value of this field should be the alias of the client certificate stored in the smkeydatabase.

**Password**

Specifies the password used by the Identity Provider or Attribute Authority to access the Service Provider or SAML Requester through the back-channel. Enter a valid string between 3 and 255 characters.

**Confirm Password**

Confirms the entry in the Password field.

## SAML 2.0 Auth. Scheme Properties Dialog--Advanced Tab

The Advanced tab is where you specify advanced configuration for the Message Extension Consumer API and optional redirect URLs for assertion processing errors during authentication.

The tab contains the following fields and controls:

**Message Consumer Plugin Group Box****Full Java Class Name**

(Optional) Specifies the fully qualified Java class name of a class which implements a Message Consumer Plug-in interface for the authentication scheme.

**Parameter**

If a value is entered in the Full Java Class Name field, this field specifies a string of parameters the API passes to the specified plug-in.

**Status Redirect URLs and Modes Group Box**

Assertion-based authentication can fail at the site that consumes assertions for various reasons. If authentication does fail, Federation Security Services provides functionality to redirect the user to different applications (URLs) for further processing. For example, when user disambiguation fails, SiteMinder can be configured to redirect the user to a provisioning system, which could create a user account based on the information found in SAML assertion.

The following options redirect the user to a configured URL based on the condition that caused the failure.

**Redirect URL for the User Not Found Status**

(Optional) Identifies the URL where SiteMinder redirects the user when the user is not found. The user not found status applies when the single sign-on message has no LoginID or the user directory does not contain the LoginID.

#### **Redirect URL for the Invalid SSO Message Status**

(Optional) Identifies the URL where SiteMinder redirects the user if one of the following conditions occur:

- The single sign-on message is not well formed, according to the XML specification
- The single sign-on message is invalid based on rules specified by the SAML schemas
- The consumer requires an encrypted assertion but the single sign-on message does not contain an encrypted assertion.

#### **Redirect URL for the Unaccepted User Credential (SSO Message) Status**

(Optional) Identifies the URL where SiteMinder redirects the user for all other error conditions other than when a user is not found or the single sign-on message is invalid. The assertion is valid, but SiteMinder does not accept the message for certain reasons, such as:

- XML digital signature validation fails
- XML decryption operation fails
- XML validation of conditions fails, such as an expired message or an audience mismatch
- None of the assertions in SSO message contain an authentication statement.

#### **Mode**

Specifies the method by which SiteMinder redirects the user to the redirect URL. The options are:

##### **302 No Data (default)**

Redirects user with an HTTP 302 redirect with a session cookie, but no other data.

##### **Http Post**

Redirects user using HTTP Post protocol.

#### **Additional URL Configuration**

Allows you to specify additional redirect URLs for HTTP status errors. The user can be redirected to a custom error page for further action.

Click this button to open the Additional URL Configuration dialog, which is described in the next topic.

## Proxy Group Box

### Server

If your network has a proxy server between the client and the system where Federation Web Services is running (that is, the system where Web Agent Option Pack is installed) specify the scheme and authority portions of the URL, such as *protocol:authority*. The scheme is http: or https: and the authority is //host.domain.com or //host.domain.com:port. For example, http://example.ca.com.

## SAML 2.0--Additional URL Configuration

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.

The Additional URL Configuration dialog lets you configure redirect URLs for error handling. This dialog includes the following settings:

### Enable Server Error URL

Specifies the URL where the user is redirected when an HTTP 500 Server error occurs. A user can encounter a 500 error because an unexpected condition prevents the web server from fulfilling the client request. If this type of error occurs, the user is sent to the specified URL for further processing.

Example: http://www.redirectmachine.com/error\_pages/server\_error.html

### Enable Invalid Request URL

Specifies the URL where the user is redirected when an HTTP 400 Bad Request or a 405 Method Not Allowed error occurs. A user can encounter a 400 error because a request is malformed or a 405 error because the web server does not allow a particular method or action to be performed. If these types of errors occur, the user is sent to the specified URL for further processing.

Example: http://www.redirectmachine.com/error\_pages/invalidreq\_error.html

### **Enable Unauthorized Access URL**

Specifies the URL where the user is redirected when an HTTP 403 Forbidden error occurs. A user can encounter a 403 error because the URL in a request is pointing to the wrong target, such as a directory instead of a file. If this error occurs, the user is sent to the specified URL for further processing.

Example:

`http://www.redirectmachine.com/error_pages/unauthorized_error.html`

### **Mode Fields**

For each URL, you can select a mode by which the browser redirects the user.

#### **302 No Data**

Redirects the user using an HTTP 302 redirect. When the user is redirected to the specified custom error page URL, they are redirected only with a session cookie and no information appended to the redirect URL.

#### **HTTP POST**

Redirects the user using the HTTP POST protocol. When the user is redirected to the custom error page URL, the following data, if available, is appended to the redirect URL and then posted to custom error page:

- TARGET
- AUTHREASON
- CONSUMERURL
- SAMLRESPONSE
- IDPID
- SPID
- PROTOCOLBINDING

# Chapter 26: SAML Affiliations Reference

---

This section contains the following topics:

[SAML Affiliation Properties Dialog](#) (see page 583)  
[SAML Affiliation Dialog Prerequisites](#) (see page 583)  
[Navigate to the SAML Affiliation Dialog](#) (see page 583)  
[SAML Affiliation Dialog Fields and Controls](#) (see page 584)  
[SAML Affiliation Dialog--Name IDs Tab](#) (see page 585)  
[SAML Affiliation Dialog--Users Tab](#) (see page 586)  
[SAML Service Providers Tab](#) (see page 588)  
[SAML Auth Schemes Tab](#) (see page 588)

## SAML Affiliation Properties Dialog

The SAML Affiliation Dialog is where you configure a SAML affiliation.

This dialog is only available if you purchased Federation Security Services.

## SAML Affiliation Dialog Prerequisites

To successfully create an affiliation, your SiteMinder administrator account must have the Manage System Objects privilege.

## Navigate to the SAML Affiliation Dialog

You can access the SAML Affiliation dialog in the Policy Server User Interface.

**Note:** When you create or modify a Policy Server object in the Policy Server User Interface, use ASCII characters. Object creation or modification with non-ASCII characters is not supported.

### To access the SAML Affiliation dialog to create a new affiliation

1. In the Object pane, click the System tab.
2. In the System tab, click SAML Affiliations.

**Note:** If the SAML Affiliations entry is not visible in the System tab, from the menu bar select View, SAML Affiliations.

3. From the menu bar select Edit, Create SAML Affiliation.

**To access the SAML Affiliation dialog to modify an existing affiliation**

1. In the Object pane, click the System tab.
2. In the System tab, click SAML Affiliations.
3. In the list pane, either:
  - Double-click the SAML affiliation you want to edit.
  - Right-click the SAML affiliation you want to edit and select Properties of SAML Affiliation from the pop-up menu.

## SAML Affiliation Dialog Fields and Controls

The SAML Affiliation dialog contains the following fields:

**Name**

Names the affiliation. This name should be unique across all affiliate domains.

**Description**

(Optional) Describes the affiliation.

**Affiliation ID**

Specifies a URI that uniquely identifies the affiliation.

**SAML Version**

Specifies the SAML version (inactive; the value defaults to 2.0, indicating that assertions sent to Service Providers that are members of this affiliation must be compliant with SAML version 2.0).

The SAML Affiliation dialog also contains the following tabs:

**Name IDs**

Specifies the name identifier for the principal to be used when corresponding with SAML Service Providers that are members of the affiliation.

**Users**

Configures the users or groups of users for the SAML Service Providers that are members of the affiliation. Configured users can be authenticated for access to Service Provider resources; the assertion generator can create SAML assertions that include entitlement information for these users.



**SAML Service Providers**

Provides a read-only listing of all SAML Service Providers that are members of the affiliation.

**SAML Auth Schemes**

Provides a read-only listing of all SAML 2.0 authentication schemes that are members of the affiliation.

**More Information:**

[Configure SAML 2.0 Affiliations At the Identity Provider](#) (see page 349)

## SAML Affiliation Dialog--Name IDs Tab

The Name IDs tab is where you configure the name ID that names the principal and is used for user disambiguation by Service Providers and Identity Providers that are members of this affiliation.

**Name ID Format**

Specifies the format of the name identifier format. For example, the name ID might take the format of an email address, such as johndoe@example.com.

Pick one of the following:

- Unspecified
- EmailAddress
- X.509 Subject Name
- Windows Domain Qualified Name
- Entity Identifier
- Persistent Identifier
- Transient Identifier

**Name ID Type Group Box**

The Name ID group box contains radio buttons that specify the Name Identifier type.

**Static**

Indicates the Name Identifier is the value in the Static Value field. Activates the Static Value field; disables other controls.

**User Attribute**

Indicates the Name Identifier is the value in the Attribute Name field. Activates the Attribute Name field; disables other controls.

**DN Attribute**

Specifies the Name Identifier as the attribute associated with a DN. Activates the User Attribute field, the DN Spec field, and the Allow Nested Groups check box; disables the Static Value field.

**Allow Nested Groups**

Enables nested groups when selecting the DN spec. Enabled if the DN Attribute Radio Button is selected.

**Name ID Fields Group Box**

Contains fields that specify information about the selected Name Identifier. The fields in this group box are context-sensitive, being determined according to the Name ID Type selection.

**Static Value**

Specifies the static text value that will be used for all name identifiers for this Service Provider.

**User Attribute**

Specifies the name of the user attribute from within a user directory, which contains the name identifier, or the attribute associated with a group or organizational unit DN.

**DN Spec**

Specifies the group or organizational unit DN to be used for obtaining the associated attribute to be used as the name identifier.

**Lookup button**

Opens the SiteMinder User Lookup dialog to locate the user group and select a DN to populate the DN Spec field.

**More Information:**

[Assign Name IDs to Affiliations](#) (see page 350)

## SAML Affiliation Dialog--Users Tab

For the entity acting only as an Identity Provider, the information on the Users tab is not relevant. You do not have to complete the fields on this tab.

If the entity is acting as an Identity Provider and a Service Provider, then the information on the Users tab is used when this system is acting as a Service Provider. As a Service Provider, it uses the information to obtain information from an incoming SAML 2.0 assertion to locate a user record and authenticate the user.

**User Disambiguation Group Box****Xpath Query**

Specifies an XPath query that the authentication scheme applies to the assertion to obtain the LoginID.

The default XPath query used when none is configured, is:

```
/Assertion/Subject/NameID/text()
```

**Example:**

To obtain the an attribute called "FirstName" from the assertion for authentication, the XPath query is:

```
/Assertion/AttributeStatement/Attribute[@Name="FirstName"]/AttributeValue/text()
```

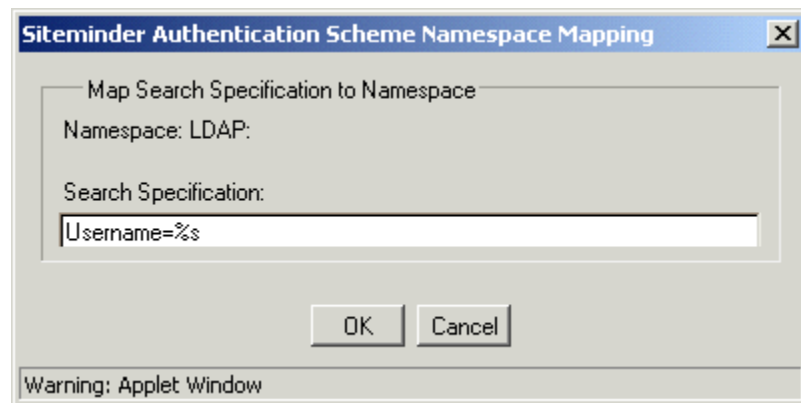
To extract the text of first Username element in the SAML assertion, use the abbreviated syntax "//Username/text()"

**Namespace**

Displays a selectable list of namespace types and defined search specifications from which you can select namespace (user directory) and then define a search specification for user disambiguation.

**Edit**

Opens the Authentication Scheme Namespace Mapping dialog where you can enter a Search Specification which defines the attribute that the authentication scheme uses to search a namespace in the Namespace list.

**SiteMinder Authentication Scheme Namespace Mapping dialog**

The Authentication Scheme Namespace Mapping dialog is where you specify the attribute that the authentication scheme uses to search a namespace.

#### **Search Specification**

Specifies the attribute that the SAML 2.0 authentication scheme uses to search a namespace. Use %s as the entry representing the LoginID.

For example, the LoginID is 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.

#### **More Information:**

[Specify Users for Disambiguation for SAML Affiliations](#) (see page 351)

## **SAML Service Providers Tab**

The SAML Service Providers tab provides a read-only list of all SAML Service Providers that belong to the affiliation. You add Service Providers to the affiliation's membership or remove them through the SAML Service Provider dialog.

#### **More Information:**

[View a List of Service Providers in an Affiliation](#) (see page 352)

## **SAML Auth Schemes Tab**

Provides a read-only listing of all SAML 2.0 authentication schemes that are members of an affiliation. You add authentication schemes to the affiliation's membership or remove them through the SAML 2.0 Auth Scheme Properties dialog.

# Chapter 27: Attribute Authority and Query Reference

---

This section contains the following topics:

[SAML 2.0 Attribute Authority Reference](#) (see page 589)

[SAML 2.0 Attribute Query Reference](#) (see page 591)

## SAML 2.0 Attribute Authority Reference

You can configure an Identity Provider to act as an Attribute Authority. The Authority provides attributes to a SAML Requester so the requester can authorize a user based on the retrieved attributes.

### Attribute Svc Dialog

You configure the Attribute Authority in the Attribute Svc tab of the SAML Service Provider Properties dialog.

**Note:** The attributes used by the Attribute Authority are configured on the [Attributes tab of the SAML Service Provider Properties dialog](#) (see page 555).

### Attribute Svc Dialog Prerequisites

To successfully configure a SAML 2.0 Attribute Authority:

- You must have installed the Policy Server and Policy Server Option Pack.
- Your SiteMinder administrator account must have the Manage System and Domain Objects privilege.

### Navigate to the Attribute Svc Dialog

**To access the Attribute Svc dialog, do the following**

1. Log on to the Policy Server User Interface.

**Note:** When you create or modify a Policy Server object in the Policy Server User Interface, use ASCII characters. Object creation or modification with non-ASCII characters is not supported.

2. Select an existing Service Provider object or create a new one.  
The SAML Service Provider Properties Dialog opens.
3. Select the Attribute Svc tab.

## Attribute Svc Dialog Fields and Controls

The following fields and controls are located on the Attribute Svc tab:

### **Enabled**

Lets the Service Provider act as an Attribute Authority, able to generate attribute assertions based on a query message from a SAML requester.

### **Validity Duration**

Specifies the number of seconds that the assertion is valid.

Default: 60 seconds

### **Require Signed Attribute Query**

Indicates that the Attribute Authority requires a digitally signed attribute query from the SAML Requester.

### **Signing Options**

Designates the Attribute Authority's signing requirements for attribute assertions and responses.

#### **Sign Assertion**

Instructs the Attribute Authority to sign only the attribute assertion. The SAML response is not signed.

#### **Sign Response**

Instructs the Attribute Authority to sign only the SAML response.

#### **Sign Both**

Instructs the Attribute Authority to sign the attribute assertion and the SAML response.

#### **Sign Neither**

Instructs the Attribute Authority not to sign the attribute assertion nor the SAML response.

**User Lookup Group Box****Namespaces**

Defines search specifications for user directory namespaces. The search specification is used by the Attribute Authority to locate the user locally using the NameID provided in the subject of the Attribute Query message.

**Edit**

Lets you access the Authentication Scheme Namespace Mapping dialog after selecting a namespace.

**Note:** At least one search specification must be provided.

**Attribute Service Namespace Mapping Dialog**

The Attribute Service Namespace Mapping dialog is where you specify the attribute that the Attribute Authority uses to search a namespace. This dialog contains the following field:

**Search Specification**

Specifies the attribute that the Attribute Authority uses to search a namespace. Use %s in the entry as a variable representing the name ID.

**More Information:**

[Use an Attribute Authority to Authorize Users](#) (see page 393)

[SAML Service Provider Dialog--Attributes Tab](#) (see page 555)

[SAML Service Provider Dialog--Encryption Tab](#) (see page 560)

## SAML 2.0 Attribute Query Reference

The SAML Requester, which in a SiteMinder environment is a Service Provider, sends an attribute query to the Attribute Authority. You configure the contents of the attribute query in the Attributes tab of a given SAML 2.0 authentication scheme.

### Attributes Dialog Prerequisites

To successfully configure an attribute query:

- You must install the Web Agent and the Web Agent Option Pack or the SPS federation gateway to use the Federation Web Services application.
- Your SiteMinder administrator account must have the Manage System and Domain Objects privilege.

## Navigate to the Attributes Dialog

To access the Attributes dialog to configure a new attribute query do the following:

1. Log on to the Policy Server User Interface.

**Note:** When you create or modify a Policy Server object in the Policy Server User Interface, use ASCII characters. Object creation or modification with non-ASCII characters is not supported.

2. Select an existing SAML 2.0 authentication scheme or create a new one.

The Authentication Scheme Properties Dialog opens.

3. Click on Additional Configuration.

The SAML 2.0 Auth Scheme Properties dialog opens.

4. Click the Attributes tab.

## Attributes Dialog for Attribute Queries

In the Attributes tab, you configure the attributes that the SAML Requester can retrieve from an Attribute Authority. These attributes will be included in the attribute query sent to the Attribute Authority.

The dialog contains the following fields and controls:

### Supported Attributes Group Box

Lists the attributes that can be requested from a SAML Attribute Authority along with the associated script stored in the policy store.

### Attribute Query Group Box

#### Enabled

Checks to generate attribute queries.

#### Sign Attribute Query

Instructs the SAML Requester to sign the attribute query prior to sending it to the Attribute Authority.

#### Require Signed Assertions

Indicates that the SAML Requester only accepts attribute assertions that are signed by the Attribute Authority. The assertion is rejected if it is not signed.



**Attribute Service**

URL of the Attribute Service at the Attribute Authority. Must be a valid URL of less than 1024 characters.

**Get All Attributes**

Indicates that the attribute query should retrieve all attributes configured at the Attribute Authority. Checking this box results in an attribute query without any attributes; however, the Attribute Authority returns all the configured attributes, that is, attributes associated with the Attribute Service retrieval method.

In some cases, checking this box avoids some performance overhead because less checking is done when constructing the query and verifying that requested attributes are still valid at the Attribute Authority. However, if there are a large number of configured attributes at the Attribute Authority and encryption is used, then checking this box might degrade performance.

**Add/Edit Attribute Dialog****Local Name**

Specifies the unique local name of the attribute used only by the authentication scheme and the federation attribute variables that you configure. The value of this field must match the Local Name field for the federation attribute variable, which is used in the policy expression that will be used when configuring the authentication scheme. Enter a unique alphanumeric string between 1 and 1024 characters.

**Note:** You can have attributes with the same name but with different name formats.

The Local Name field enables you to change the name of the requested attribute without changing the policy expression that protects the resource.

Example:

Attribute configuration:

- Local Name:local1
- Attribute Name: phone
- Name Format:unspecified

Federation Attribute Variable configuration:

- Name=fedvar1
- IdP ID=AttributeAuthority1
- Local Name=local1

Policy expression:

- (fedvar1=="STATIC\_VALUE")

If you change only the Attribute Name from **phone** to **title**, only the requested attribute changes; the federation variable and the policy expression do not have to change.

#### Attribute Name

Specifies the unique name that the SAML requester includes in the attribute query. This value must match the value of the Variable Name field specified for an attribute configured at the Attribute Authority.

#### Name Format

Defines the format of the attribute. The options are:

- unspecified
- basic
- uri

Refer to the SAML 2.0 specification for definitions of these formats.

#### More Information:

[Enable Attribute Queries and Specify Attributes](#) (see page 401)

## Name IDs Dialog Fields for Attribute Queries

The Name IDs tab is where you configure the Name ID included in the attribute query sent from the SAML Requester to the Attribute Authority.

This tab is only configurable if the Attribute Query feature is enabled, that is, the Enabled checkbox on the Attributes tab is checked.

The fields on this dialog are as follows:

#### Name ID Format

Specifies the format of the Name ID. This value must match the format of the expected Name ID at the Attribute Authority or the request will fail.

**Name ID Type**

This field defines the type of attribute used for the Name Identifier.

**Static**

Indicates that the Name Identifier is the value in the Static Value field.  
Activates the Static Value field; disables other controls.

**User Attribute**

Indicates that the Name Identifier is the value in the Attribute Name field.  
Activates the Attribute Name field; disables other controls.

**DN Attribute**

Indicates that the Name Identifier is an attribute associated with a DN.  
Activates the User Attribute field, the DN Spec field, and the Allow Nested Groups check box; disables the Static Value field.

**Allow Nested Groups**

Indicates that nested groups are allowed when selecting the DN. Enabled if the DN Attribute Radio Button is selected.

**Name ID Fields Group Box**

Contains fields that specify information about the selected Name Identifier. The fields in this group box are context-sensitive, being determined according to the Name ID Type selection.

**Static Value**

Specifies the static text value that will be used for all name identifiers for this Service Provider.

**User Attribute**

Specifies the name of the user attribute which contains the name identifier, or the attribute associated with a group or organizational unit DN.

**DN Spec**

Specifies the group or organizational unit DN to be used for obtaining the associated attribute to be used as the name identifier.

**More Information:**

[Configure the NameID for the Attribute Query](#) (see page 402)

## Federation Attribute Variable Editor Reference

You can include a federation attribute variable for inclusion in a policy expression that protects a resource at a SAML Requester site.

The fields for the Attribute Variable Editor dialog are as follows:

**Name**

Names the variable.

**Description**

Provides a way to describe the variable.

**Return Type**

Specifies the type of data used to express the variable supported by the Policy Server. Options are:

- Boolean
- Date
- Number
- String

**IdP ID**

Using the IdP ID value as a reference, this parameter identifies the SAML 2.0 authentication scheme that contains the configuration for this federation attribute. When you configure a SAML 2.0 authentication scheme, IdP ID is one of the fields you set. This field for the federation attribute variable lists all IdP ID values for all configured SAML 2.0 authentication schemes. Select the IdP ID associated with authentication scheme that defines the attribute you want to use.

**Local Name**

Unique name that identifies the variable to the SAML 2.0 authentication scheme. This value must match the value you entered when adding an attribute in the Attributes tab of the SAML 2.0 Authentication Scheme.

**More Information:**

[Create a Federation Attribute Variable](#) (see page 403)

# Chapter 28: WS-Federation Resource Partner Properties Reference

---

This section contains the following topics:

[Resource Partner Properties Dialog](#) (see page 597)  
[Resource Partner Properties Dialog Prerequisites](#) (see page 597)  
[Navigate to the Resource Partner Properties Dialog](#) (see page 598)  
[Resource Partner Properties Dialog Fields and Controls](#) (see page 598)  
[Resource Partner Properties Dialog--Users Tab](#) (see page 601)  
[Resource Partner Properties Dialog--Name IDs Tab](#) (see page 602)  
[Resource Partner Properties Dialog--General Tab](#) (see page 603)  
[Resource Partner Properties Dialog--SSO Tab](#) (see page 604)  
[Resource Partner Properties Dialog--Attributes Tab](#) (see page 607)  
[Resource Partner Properties Dialog--Signout Tab](#) (see page 609)  
[Resource Partner Properties Dialog--Advanced Tab](#) (see page 610)

## Resource Partner Properties Dialog

If you have purchased SiteMinder Federation Security Services, you can use the Resource Partner Properties Dialog to configure a WS-Federation Resource Partner.

### More Information:

[Resource Partner Properties Dialog Fields and Controls](#) (see page 598)

## Resource Partner Properties Dialog Prerequisites

The following prerequisites must be met to successfully establish an affiliate domain with a Resource Partner:

- Your SiteMinder administrator account must have the Manage Domain Objects privilege.
- The Affiliate Domain in which the Resource Partner is to be configured must pre-exist.
- Federation Web Services must be installed on a system.  
Federation Web Services can be installed by the Web Agent Option Pack or by the SPS federation gateway.
- The assertion generated at the Account Partner must be forwarded to an application at the Resource Partner that can receive and interpret it.

## Navigate to the Resource Partner Properties Dialog

You can access the Resource Partner Properties dialog in the Policy Server User Interface.

**Note:** When you create or modify a Policy Server object in the Policy Server User Interface, use ASCII characters. Object creation or modification with non-ASCII characters is not supported.

### To access the Resource Partner Properties dialog

1. If necessary, in the Object pane, click on the Domains tab.
2. In the Domains tab, click the affiliate domain to which you want to add a Resource Partner.
3. From the menu bar, select:  
Edit *affiliate\_domain\_name*, Create Resource Partner

### To access the Resource Partner Properties dialog to edit an existing object

1. If necessary, click on the Domains tab.
2. Expand the affiliate domain that contains the Resource Partner that you want to edit.
3. Click on the Resource Partners entry.
4. In the List pane, click the partner that you want to edit.
5. From the menu bar, select:  
Edit, Properties of Resource Partner

## Resource Partner Properties Dialog Fields and Controls

The Resource Partner Properties dialog contains the following fields and controls:

### Name

Name of the Resource Partner. This name should be unique across all affiliate domains.

### Description

(Optional) a brief description of the Resource Partner.

**Authentication URL**

Defines the protected URL used for authentication purposes when a user tries to access Resource Partner resource but does not have a SiteMinder session. When a user who has not logged in at the Account Partner requests a protected Resource Partner resource, the user is sent to the 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.

**Important!** You must create a policy to protect the Authentication URL; however, do this after you add a Resource Partner to an affiliate domain because you have to exit the Resource Partner Properties dialog.

**Enabled**

This check box must be marked for the Policy Server and Federation Web Services to support authentication of Resource Partner resources.

**Use Secure URL**

The Use Secure URL feature instructs SiteMinder's Single Sign-on Service to encrypt only the SMPORTALURL query parameter that it appends to the Authentication URL prior to redirecting the user to establish a SiteMinder session. After the user is authenticated, the Authentication URL redirects the user back to the destination specified by the SMPORTALURL query parameter.

Encrypting the SMPORTALURL prevents a malicious user from modifying the value and redirecting authenticated users to a malicious web site instead of the originally requested target.

If you select this checkbox, you must do the following:

1. Set the Authentication URL field to the following URL:

`http(s)://idp_server:port/affwebservices/secure/securedirect`

2. Protect the securedirect web service with a SiteMinder policy.

If the SiteMinder Producer/Identity Provider/Account Partner serves more than one Consumer/Service Provider/Resource Partner, it probably authenticates different users for these different partners. As a result, for each Authentication URL that uses securedirect, this web service must be included in a different realm for each partner.

To associate the securedirect service with different realms, modify the web.xml file to create different resource mappings; you cannot copy the securedirect web service to different locations on your server. This file is located in the directory `web_agent_home/affwebservices/WEB-INF`.

### **Application URL**

(Optional) Protected URL for a custom web application that is used to supply user attributes to the SiteMinder Single Sign-on service. The application can be on any host in your network.

Attributes from the web application specified in this field are made available to the Assertion Generator and then placed in the SAML assertion by an Assertion Generator plug-in. You must write and integrate the plug-in with SiteMinder.

The FWS application supplies the sample web applications that you can use as a basis for your web application. These applications are located as follows:

`http://ap_server:port/affwebservices/public/sample_application.jsp`

`http://ap_server:port/affwebservices/public/unsolicited_application.jsp`

### **ap\_server:port**

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

The Resource Partner Properties dialog also contains the following tabs:

### **Users tab**

Configures the users or groups of users that can be authenticated for access to Resource Partner resources; the assertion generator can create SAML assertions that include information about these users.

### **Name IDs tab**

Specifies the Name Identifier to be used when corresponding with this Resource Partner.

### **General tab**

Specifies general information about the Resource Partner.

### **SSO tab**

Configures the Single Sign-On (SSO) profile of a Resource Partner.

### **Attributes tab**

Configures assertion attribute statements, which define the user attributes, DN attributes, or static data that are passed from the Account Partner to the Resource Partner in assertions.

### **Signout tab**

Configures the WS-federation signout feature.

### **Advanced tab**

(Optional) Configures an Assertion Generator plug-in.



## Resource Partner Properties Dialog--Users Tab

The Users tab is where you specify the users and groups that can be granted access to resources at the Resource Partner. The Assertion Generator at the Account Partner can generate assertions for these users for authentication purposes.

You can only add users and groups from directories that are included in the affiliate domain in which the Resource Partner exists.

In this dialog, you will find the following fields and controls:

### **Add/Remove**

Opens the Users/Groups dialog, where you can add and remove users and groups to the Current Members list box.

### **Users/Groups**

Lists the users and groups that are included or excluded by the policy that governs assertion production for a particular user directory. The list box contains one tab for each user directory in the affiliate domain.

### **AND Users/Groups**

Lets you configure a policy with an OR or AND relationship between the users or groups of users to instruct the Policy Server when to fire the rules included in a policy. For example, you could define the following group relationship in a policy:

Authorize if Member of (Accounting AND Marketing AND Collections) OR (Legal AND Finance AND Support) OR (Shipping)

In this example, the Policy Server would only fire the policy when the user is a member of all the groups in the AND relationship or a member of the groups in the OR relationship.

### **More Information:**

[Select Users for Which Assertions Will Be Generated](#) (see page 407)

## Resource Partner Properties Dialog--Name IDs Tab

The Name IDs tab is where you configure the Name Identifier, which names a user in a unique way in the SAML assertion. The format of the Name Identifier establishes the type of content used for the ID. For example, the format might be the User DN, in which case the content might be a uid.

### Name ID Format

Specifies the format for the Name ID. Pick one of the following options:

- EmailAddress
- UPN
- Common Name

For a description of each format, see *Assertions and Protocols for the OASIS Security Assertion Markup Language (SAML) V2.0* specification (sstc-saml-core-2.0-cd-04.pdf).

### Name ID Type Group Box

The Name ID group box contains radio buttons that specify the Name Identifier type. The selections are:

#### Static

Indicates that the Name Identifier is the value in the Static Value field. Activates the Static Value field; disables other controls.

#### User Attribute

Indicates the Name Identifier is specified in the Attribute Name field. Activates the Attribute Name field; disables other controls.

#### DN Attribute

Indicates the Name Identifier is specified by an attribute associated with a DN. Activates the User Attribute field, the DN Spec field, and the Allow Nested Groups check box; disables the Static Value field.

#### Allow Nested Groups

Indicates that nested groups are allowed when selecting the DN. Enabled if the DN Attribute Radio Button is selected.

**Name ID Fields Group Box**

Contains fields that specify information about the selected Name Identifier. The fields in this group box are context-sensitive, being determined according to the Name ID Type selection.

**Static Value**

Specifies the static text value that will be used for all name identifiers for this Service Provider.

**User Attribute**

Specifies the name of the user attribute which contains the name identifier, or the attribute associated with a group or organizational unit DN.

**DN Spec**

Specifies the group or organizational unit DN to be used for obtaining the associated attribute to be used as the name identifier.

**Lookup button**

Opens the SiteMinder User Lookup dialog to locate the user group and select a DN to populate the DN Spec field.

**More Information:**

[Specify Name IDs for WS-Federation Assertions](#) (see page 410)

## Resource Partner Properties Dialog--General Tab

The General tab is where you specify identity information for the Account Partner and Resource Partner other general information.

**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.

### D-Sig Verification Info Group Box

Determines digital signature functions.

#### Disable Signature Processing

If set, all signature processing for this Resource Partner (both signing and verification of signatures) is disabled for the Resource Partner.

**Note:** Signature processing *must* be enabled in a production environment. Disabling it by setting the Disable Signature Processing option should be used for debugging only.

#### Signing Alias

(Optional) Specifies the alias associated with a specific private key in the SiteMinder key database. By completing this field, you are indicating which private key the Account Partner should use to sign assertions and assertion responses.

**Note:** It is recommended that the private key already be stored in the key database before you specify its associated alias in this field.

**Limits:** an alphanumeric character string

### More Information

[Configure Required General Information for WS-Federation](#) (see page 411)

## Resource Partner Properties Dialog--SSO Tab

The SSO tab is where you configure Single Sign-On (SSO) information for the WS-Federation Resource Partner.

### Authentication Method

Specifies the URI for the AuthenticationMethod attribute placed in the SAML assertion. Specify a an authentication method as dictated by the SAML specification and one that is appropriate for the Authentication Level specified for the Resource Partner.

For example, if the authentication level is certificate plus basic, the default Authentication Level is 15. The default value is Password. For this default value, the URI would in the assertion would be:

`urn:oasis:names:tc:SAML:1.0:am:classes:password.`

**Validity Duration**

Specifies a number of seconds (a positive integer) for which a generated assertion is valid. The default is 60 seconds.

In a test environment, you may want to 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 2006) is after SessionNotOnOrAfter time (Fri Sep 09 17:28:20 EDT 2006)
```

**Security Token Consumer Service**

Specifies the URL of the service at the Resource Partner that receives security token response messages and extracts the assertion. The default location for the SiteMinder service is:

`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. These components provide the Federation Web Services application.

**Note:** The WSFedDispatcher Service receives all incoming WS-Federation messages and forwards the request processing to the appropriate serviced based on the query parameter data. Although there is a wsfedsecuritytokenconsumer service, the wsfeddispatcher service is recommended for the entry in this field.

**Authentication Level**

Specifies the minimum level at which the user must have authenticated to gain access to a SiteMinder realm. If the user has authenticated at this level, the Account Partner will generate an assertion for the user. If the user is not authenticated at this level, they are redirected to the Authentication URL to authenticate at this level.

**Note:** The user must have authenticated at this level or greater.

**Restrictions**

Opens the Restrictions dialog from where you can configure IP address and time restrictions on the assertion generation policy.

**More Information:**

[Restrictions Dialog for a Resource Partner](#) (see page 606)  
[Configure Single Sign-on for WS-Federation](#) (see page 412)

## Restrictions Dialog for a Resource Partner

The Restrictions dialog is where you configure IP address and time restrictions for the assertion generation policy for the Resource Partner.

## Restrictions Dialog Prerequisites

You must be in the process of configuring Single Sign-On for a Resource Partner from the Resource Partner Properties dialog.

## Restrictions Dialog Field and Controls

The IP Addresses group box on the Restrictions dialog contains the following fields:

### IP Addresses

A list of restricted IP addresses configured for the policy for the Resource Partner resources. The Policy Server User Interface 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.

### Add

Opens an empty Add IP Address dialog from where you can create a new IP address restriction.

### Edit

Opens the IP restriction entry selected in the IP Addresses list for editing in the Add IP Address dialog.

### Remove

Deletes the IP restriction entry selected in the IP Addresses list.

The Time Restrictions Group Box contains the following field:

### Set

Opens the Time dialog so you can configure the availability of the Resource Partner's availability. When you add a time restriction, the Resource Partner functions only during the period specified.

### More Information:

[Specify IP Address Restrictions for Resource Partners \(optional\)](#) (see page 415)  
[Set up Time Restrictions for Resource Partner Availability \(optional\)](#) (see page 416)

## Resource Partner Properties Dialog--Attributes Tab

The Attributes tab is where you configure SAML assertion attribute statements. You also indicate whether the attribute will be used for calls to the Attribute Authority or for single sign-on.

The controls on this tab are as follows:

**Attribute List**

Selectable list of configured attributes.

**Create**

Opens the Service Provider Attribute Editor dialog from where you can configure Service Provider entitlement attributes.

**Edit**

Opens the attribute entry that you selected in the Attribute List for editing.

**Remove**

Deletes the attribute entry selected in the Attribute List.

**Allow Nested Groups**

Allows nested groups to be searched in an LDAP Directory.

## Resource Partner Attribute Dialog Prerequisites

You must be in the process of configuring a WS-Federation Resource Partner from the Resource Partner Properties dialog.

## Resource Partner Attribute Dialog--Attribute Setup Tab

The Attribute Setup tab contains the following fields and controls:

**Attribute**

Specify the format for the attribute that will be part of a SAML assertion. Options are:

- EmailAddress
- UPN
- CommonName
- Group
- NameValue

### **Attribute Kind**

The Attribute Kind group box contains radio buttons that allow you to specify the type of attribute that will be added to the assertion:

#### **Static**

Indicates that the attribute kind is the value in the Variable Name field. Activates the Variable Value field; disables other controls in the dialog.

#### **User Attribute**

Indicates that the DN identifier is the value of Attribute Name field. Activates the Attribute Name field; disables other controls in the dialog.

#### **DN Attribute**

Indicates that the name identifier will be specified by an attribute associated with a DN. Activates the User Attribute field, the DN Spec field, and the Allow Nested Groups field, and disables the Static Value field.

Note: 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.

#### **Allow Nested Groups**

Indicates that nested groups are allowed when selecting the DN. Enabled if the DN Attribute Radio Button is selected.

Note: If the attribute is configured to be encrypted, the Encryption Certificate settings in the Encryption tab will be enabled.

### **Attribute Fields**

Complete the fields for the Attribute Kind that you selected. Different fields become available depending on the attribute type (static, user attribute, DN attribute).

#### **Variable Name**

Designates the name for the attribute that SiteMinder returns in the assertion. Complete this field for any attribute type.

#### **Variable Value**

(Static only) Defines the static text as the value for the name/value pair. Enter a value.

#### **Attribute Name**

(User and DN Attributes only) Specifies the user directory attribute SiteMinder uses for the name/value pair.

#### **DN Spec**

(DN Attribute only) Specifies the DN of the user or user group and the name of the user attribute in the Attribute Name field.



**More Information:**

[Configure Attributes for WS-Federation Assertions \(optional\)](#) (see page 417)

## Resource Partner Attribute Dialog--Advanced Tab

The Advanced tab of the Resource Partner Attribute dialog contains the following field:

**Script**

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.

**Note:** If you copy and paste the contents of the Script field for another entitlement, you must select the appropriate radio button in the Attribute Kind group box of the Attribute Setup tab.

## Resource Partner Properties Dialog--Signout Tab

The Signout tab lets you enable the WS-Federation signout feature, which terminates all sessions for a particular user at each Resource Partner where the user has a session.

The fields and controls on this dialog are as follows:

**Enable Signout**

Enables the WS-Federation signout feature for the Resource Partner.

**SignoutCleanupURL**

Specifies a URL at the Resource Partner where the user's browser is redirected to terminate the user session.

Federation Web Services redirects the user to the signout cleanup URL after the user's session is completely removed at the Account Partner and all Resource Partner sites. If the signout is initiated at the Resource Partner, the logout confirmation page should be an unprotected resource at the Resource Partner site. If signout is initiated at an Account Partner site, the logout confirmation page should be an unprotected resource at the Account Partner site.

**SignoutConfirmURL**

Specifies the URL at the Account Provider where the signout message and Resource Provider cleanup locations are sent as post data to the SignoutConfirm JSP page.

The default URL is:

`http://ap_server:port/affwebservices/public/signoutconfirmurl.jsp`

**ap\_server:port**

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

The signoutconfirmurl.jsp is included with the Web Agent Option Pack or SPS federation gateway. If you move this page from the default directory, make sure you put it in a location where it is accessible to the servlet engine you are using for the Federation Web Services application.

**More Information:**

[Configure Signout](#) (see page 422)

## Resource Partner Properties Dialog--Advanced Tab

The Advanced tab is where you can configure a custom-developed Assertion Generator plugin developed using the Assertion Generator Plugin API. This task is optional.

The Assertion Generator Plugin Group Box has the following fields:

**Full Java Class Name**

Specifies the fully qualified Java class name of the Assertion Generator Plugin.

**Parameters**

Specifies a string of parameters to be passed to the specified plugin when the Full Java Class Name field is configured.

# Chapter 29: WS-Federation Authentication Scheme Reference

---

This section contains the following topics:

[Authentication Scheme Properties Dialog--WS-Federation Template](#) (see page 611)  
[WS-Federation Auth. Scheme Properties Dialog Prerequisites](#) (see page 612)  
[Navigate to the Auth. Scheme Properties Dialog--WS Federation Template](#) (see page 612)  
[Federation Authentication Scheme Dialog--Scheme Setup Tab](#) (see page 613)  
[WS-Federation Authentication Scheme Dialog--Advanced Tab](#) (see page 614)  
[WS-Federation Auth. Scheme Fields and Controls](#) (see page 615)  
[Auth. Scheme Properties--WS-Federation--Users Tab](#) (see page 616)  
[Auth. Scheme Properties--WS-Federation--SSO Tab](#) (see page 617)  
[Auth. Scheme Properties--WS-Federation--Signout Tab](#) (see page 619)  
[Auth. Scheme Properties--WS-Federation--Advanced Tab](#) (see page 620)

## Authentication Scheme Properties Dialog--WS-Federation Template

If you have purchased SiteMinder Federation Security Services, you can configure the WS-Federation authentication scheme. After the scheme is configured, you can assign it to a realm.

### Scheme Common Setup Group Box

This group box lets you identify the authentication scheme.

#### Name

Designates a name for the authentication scheme.

**Note:** When Active Directory Application Mode (ADAM) is used as a policy store, the maximum length for the authentication scheme name is 22 characters.

#### Description

Provides a description of the authentication scheme.

#### Protection Level

Allows single sign-on for authentication schemes of equal or lower protection levels within the same policy domain, while requiring additional authentication to access resources with higher protection level schemes.

**Limits:** 1 and 1000.

Authentication schemes have a default protection level that you may need to change. High protection levels should be used for critical resources, while lower level schemes should be used for commonly accessible resources.

#### More Information:

[Federation Authentication Scheme Dialog--Scheme Setup Tab](#) (see page 613)

[WS-Federation Authentication Scheme Dialog--Advanced Tab](#) (see page 614)

## WS-Federation Auth. Scheme Properties Dialog Prerequisites

To successfully configure a WS-Federation authentication scheme:

- You must have installed the Policy Server and Policy Server Option Pack.
- Your SiteMinder administrator account must have the Manage System and Domain Objects privilege.

## Navigate to the Auth. Scheme Properties Dialog--WS Federation Template

You can access the Authentication Scheme dialog in the Policy Server User Interface.

**Note:** When you create or modify a Policy Server object in the Policy Server User Interface, use ASCII characters. Object creation or modification with non-ASCII characters is not supported.

To access the Authentication Scheme Dialog for a new user directory connection, do the following:

1. Begin at the menu bar.
2. Select Edit, System Configuration, Create Authentication Scheme

The Authentication Scheme Dialog opens.

To access the Authentication Scheme Dialog for an existing authentication scheme:

1. In the object pane, click on Authentication Schemes.
2. In the Authentication Scheme List pane, select an existing authentication scheme.
3. Right click and select Properties of Authentication Scheme.

## Federation Authentication Scheme Dialog--Scheme Setup Tab

The Scheme Setup tab for WS-Federation authentication is where you specify how the Resource Partner communicates with the Account Partner to retrieve the assertion, authenticate a user based on that assertion, then direct the user to the target resource.

The fields in the dialog box are as follows:

### Resource Partner ID

Specifies a URI that uniquely identifies the Resource Partner.

**Note:** The value that you enter must match the value of the ID specified for the corresponding Resource Partner object that you establish at the Account Partner.

### Account Partner ID

Specifies a URI that uniquely identifies the Account Partner from which assertions for this Resource Partner are issued.

The Resource Partner will accept assertions only from this Account Partner.

**Note:** The value that you enter for the issuer must match the value of the Account Partner ID configured at the Account Partner site.

### Skew Time

Determines 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.

### **D-Sig Verification Group Box**

Contains fields and controls that allow you to specify signature verification information.

#### **Disable Signature Processing**

Disables all signature processing, that is, signing and verification of signatures, for this Resource Partner.

**Caution:** Signature processing must be enabled in a production environment. Disabling it by setting the Disable Signature Processing option should be used for debugging only.

#### **Alias**

Specifies a unique string of only alphanumeric characters, which locates the certificate in the Resource Partner's smkeydatabase used to verify the digital signature of the assertion. This certificate is associated with the private key stored at the Account Partner. An alias entry is required for signature verification.

#### **Additional Configuration**

Opens the WS-Fed Auth Scheme Properties dialog. From this dialog, you can specify additional configuration details for the authentication scheme.

#### **More Information:**

[Configure the WS-Federation Authentication Scheme](#) (see page 434)

## **WS-Federation Authentication Scheme Dialog--Advanced Tab**

The Advanced tab of the Authentication Scheme Properties dialog for WS-Federation is where you define information about the shared library that processes WS-Federation authentication.

The fields on the Advanced tab are as follows:

#### **Library**

Designates 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 for WS-Federation authentication is smauthsaml.

#### **Parameter**

Displays the information you entered in the Scheme Setup tab.

## WS-Federation Auth. Scheme Fields and Controls

When you click on the Additional Configuration button from the main Authentication Scheme Properties window, the WS-Federation Auth. Scheme Properties dialog is displayed.

The WS-Federation Auth. Scheme Properties dialog contains the following tabs:

### **Users**

Specifies how to obtain the user information from an assertion in an incoming <RequestSecurityTokenResponse> message for authentication purposes.

### **SSO**

Defines the single sign-on bindings supported by the Resource Partner.

### **Signout**

Enables the Resource Partner to respond to a signout request sent from an Account Partner.

### **Advanced**

Defines advanced configuration for the Message Extension Consumer API and optional redirect URLs for assertion processing errors.

## Auth. Scheme Properties--WS-Federation--Users Tab

The Users tab defines how to obtain user information from an assertion in an incoming <RequestSecurityTokenResponse> message.

The tab includes the following fields and controls:

### User Disambiguation Group Box

#### Xpath Query

XPath query that the authentication scheme applies to the assertion to obtain the LoginID.

The default XPath query used when none is configured, is:

```
/Assertion/Subject/NameID/text()
```

Xpath queries should not contain namespace prefixes. The following 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()
```

#### Example

To obtain the attribute called "FirstName" from the assertion for authentication, the XPath query is:

```
/Assertion/AttributeStatement/Attribute[@Name="FirstName"]/AttributeValue/text()
```

#### Namespace

Selectable List of namespace types and defined search specifications from which you can select namespace (user directory) type and then define a search specification for user disambiguation.

#### Edit

Opens the Authentication Scheme Namespace Mapping dialog where you can enter a Search Specification that defines the attribute that the authentication scheme uses to search a namespace. Use %s as the entry representing the LoginID.

For example, the LoginID is 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.



### Authentication Scheme Namespace Mapping Dialog

The Authentication Scheme Namespace Mapping dialog is where you specify the attribute that the authentication scheme uses to search a namespace. The dialog contains the following field:

#### Search Specification

Specifies the attribute that the WS-Federation authentication scheme uses to search a namespace. Use %s in the entry as a variable representing the LoginID.

#### More Information:

[Locate User Records for Authentication](#) (see page 436)

## Auth. Scheme Properties--WS-Federation--SSO Tab

The SSO tab is where you define how the Resource Partner processes single sign-on requests.

The tab contains the following fields:

#### Redirect Mode

Indicates the method by which the Resource Partner redirects the user to the target resource. If you select 302 No Data or 302 Cookie Data, no other configuration is required. If you select Server Redirect or PersistAttributes, there is additional configuration required.

##### 302 No Data

(Default). User is redirected via an HTTP 302 redirect with a session cookie, but no other data.

##### 302 Cookie Data

User is redirected via an HTTP 302 redirect with a session cookie and additional cookie data configured for the Resource Partner at the Account Partner.

#### Server Redirect

Enables header and cookie attribute information, which is received as part of a SAML assertion, to be passed to the custom target application. The service that collects the credentials (SAML 2.0 Assertion Consumer Service or WS-Federation Security Token Consumer Service) transfers the user to the target application URL by using server-side redirect technology. Server-side redirects are part of the Java Servlet specification, and are supported by all the standard-compliant servlet containers.

To use this mode, you must follow these requirements:

- The URL you specify for this mode must be relative to the context of the servlet that is consuming the assertion, which is typically `/affwebservices/public/`. The root of the context is the root of the Federation Web Services application, typically `/affwebservices/`.

All target application files need to be in the application's root directory. This directory is either:

—Web Agent: `web_agent_home\webagent\affwebservices`

—SPS federation gateway:  
`sps_home\secure-proxy\Tomcat\webapps\affwebservices`

- You need to define realms, rules, and policies to protect target resources. The realms must be defined with at least the value `/affwebservices/` in the resource filter.
- You must have a custom Java or JSP application on the server that is serving the Federation Web Services application—that is, the server where the Web Agent Option Pack or SPS federation gateway is installed.

Java servlet technology allows applications to pass information between two resource requests using the `setAttribute` method of the `ServletRequest` interface.

The service that consumes assertions sends the user attribute to the target application by an attribute object in the request before redirecting the user to the target application. The service that consumes assertions sends the attributes by creating a `java.util.HashMap` object. The attribute that contains the `HashMap` of SAML attributes is "Netegrity.AttributeInfo."

Two other `Java.lang.String` attributes are set by the service that consumes assertions to pass the user identity to the custom application:

—`Netegrity.smSessionID` attribute represents the SiteMinder session ID

—`Netegrity.userDN` attribute represents the SiteMinder user DN.

The custom target application at the customer site can read these objects from the HTTP request object and can make use of the data found in the `hashmap` objects.

### PersistAttributes

User is redirected via an HTTP 302 redirect with a session cookie, but no other data. Additionally, this mode instructs the Policy Server to store attributes extracted from an assertion in the session store so they can be supplied as HTTP header variables. For additional configuration, see the instructions for using [SAML attributes as HTTP headers](#) (see page 288).

**Note:** If you choose `PersistAttributes` and the assertion contains attributes that are left blank, a value of `NULL` is written to the session store. This value acts as a placeholder for the empty attribute and it is passed to any application using the attribute.

**Target**

Specifies the target resource URI at the Service Provider destination site.

**Enforce Single Use Policy**

Ensures the single use policy is enforced, selecting this option prevents assertions from being re-used at a Resource Partner to establish a second session.

**More Information:**

[Configure WS-Federation Single Sign-on at the Resource Partner](#) (see page 438)

## Auth. Scheme Properties--WS-Federation--Signout Tab

The Signout tab lets you configure the Resource Partner to respond to a signout request sent from an Account Partner.

The fields and controls on this dialog are as follows:

**Enable Signout**

Enables WS-Federation signout.

**SignoutURL**

If signout is enabled, specifies the URL of the signout servlet at the Account Partner. The following default URL is the recommended entry:

`https://<ap_service:port>/affwebservices/public/wsfeddispatcher`

**Note:** The WSFedDispatcher Service receives all incoming WS-Federation messages and forwards the request processing to the appropriate service based on the query parameter data. Although there is a wsfedsignout service, use the wsfeddispatcher URL for the SignoutURL.

**More Information:**

[Implement WS-Federation Signout](#) (see page 440)

## Auth. Scheme Properties--WS-Federation--Advanced Tab

The Advanced tab is where you specify advanced configuration for the Message Extension Consumer API and optional redirect URLs for assertion processing errors during authentication.

The tab contains the following fields and controls:

### Message Consumer Plugin Group Box

#### Full Java Class Name

(Optional) Specifies the fully qualified Java class name of a class which implements a Message Consumer Plug-in interface for the authentication scheme.

#### Parameter

If a value is entered in the Full Java Class Name field, this field specifies a string of parameters the API passes to the specified plug-in.

### Status Redirect URLs and Modes Group Box

Assertion-based authentication can fail at the site that consumes assertions for various reasons. If authentication does fail, Federation Security Services provides functionality to redirect the user to different applications (URLs) for further processing. For example, when user disambiguation fails, SiteMinder can be configured to redirect the user to a provisioning system, which could create a user account based on the information found in SAML assertion.

The following options redirect the user to a configured URL based on the condition that caused the failure.

#### Redirect URL for the User Not Found Status

(Optional) Identifies the URL where SiteMinder redirects the user when the user is not found. The user not found status applies when the single sign-on message has no LoginID or the user directory does not contain the LoginID.

#### Redirect URL for the Invalid SSO Message Status

(Optional) Identifies the URL where SiteMinder redirects the user if one of the following conditions occur:

- The single sign-on message is not well formed, according to the XML specification
- The single sign-on message is invalid based on rules specified by the SAML schemas
- The consumer requires an encrypted assertion but the single sign-on message does not contain an encrypted assertion.

**Redirect URL for the Unaccepted User Credential (SSO Message) Status**

(Optional) Identifies the URL where SiteMinder redirects the user for all other error conditions other than when a user is not found or the single sign-on message is invalid. The assertion is valid, but SiteMinder does not accept the message for certain reasons, such as:

- XML digital signature validation fails
- XML decryption operation fails
- XML validation of conditions fails, such as an expired message or an audience mismatch
- None of the assertions in SSO message contain an authentication statement.

**Mode**

Specifies the method by which SiteMinder redirects the user to the redirect URL. The options are:

**302 No Data (default)**

Redirects user with an HTTP 302 redirect with a session cookie, but no other data.

**Http Post**

Redirects user using HTTP Post protocol.



# Chapter 30: Federation Web Services URLs Used in SiteMinder Configuration

---

This section contains the following topics:

[Federation Services URLs](#) (see page 623)

[URLs for Services at the Producing Authority](#) (see page 623)

[URLs for Services at the Consuming Authority](#) (see page 633)

[The Web.xml File](#) (see page 640)

## 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 via the Policy Server User Interface, 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 Policy Server User Interface 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. It lists servlets and URL mappings.

## URLs for Services at the Producing Authority

The Federation Web Services application supplies the following services:

- [Intersite Transfer Service](#) (see page 624) (SAML 1.x producer)
- [Assertion Retrieval Service](#) (see page 625) (SAML 1.x producer)
- [Artifact Resolution Service](#) (see page 626) (SAML 2.0 IdP)
- [Single sign-on Service](#) (see page 627) (SAML 2.0 IdP)
- [Single logout Service](#) (see page 628) (SAML 2.0 IdP)

- [Identity Provider Discovery Profile Service](#) (see page 629) (SAML 2.0)
- [Attribute Service](#) (see page 629) (SAML 2.0)
- [Single sign-on Service](#) (see page 632) (WS-Federation)
- [Signout Service](#) (see page 631) (WS-Federation)
- [WSFedDispatcher Service](#) (see page 633) (WS-Federation)

Although these services are provided at the Identity Provider/Producer, you may be entering the URL for the service at the Service Provider.

## Intersite Transfer Service (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 site to a consumer site.

- 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.

- URL Entered: include URL in a hard-coded link on a page at the producer
- Associated Servlet and Servlet Mapping in 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. You have to 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 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>
```

## 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.IPDService</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>
```

## 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 that 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>ws fedssso</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>ws fedssso</servlet-name>
<url-pattern>/public/wsfedsso/*</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 that 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

```
<servlet>
  <servlet-name>wsfedsignout</servlet-name>
  <display-name>WS-Federation Signout Service</display-name>
  <description>This servlet is the WS-Federation Signout service
    at an AP.</description>
  <servlet-class>com.netegrity.affiliateminder.webservices.wsfed.
    SignoutService</servlet-class>
</servlet>

<servlet-mapping>
  <servlet-name>ws fedsignout</servlet-name>
  <url-pattern>/public/wsfedsignout/*</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 that 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 Consuming Authority

The following services are provided by the SiteMinder consuming authority:

- 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)

Although these services are provided at the consuming authority, you may be entering the URL for the service at the producing authority.

## 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 that is hosting the Web Agent Option Pack or the SPS federation gateway.

- Field Where URL Entered: link in an application at the Service Provider

This link initiates single sign-on. It should 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 that is 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>ws fedsecuritytokenconsumer</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 that is 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>wsfedsignout</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 should not need to 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 31: Troubleshooting

---

This section contains the following topics:

[General Issues](#) (see page 641)

[SAML 1.x-Only Issues](#) (see page 646)

[SAML 2.0-Only Issues](#) (see page 647)

## 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 "Java Agent API initialization FAILED" or "unsatisfied link error" messages on a system with other CA products, such as IdentityManager or SOA Security Manager. 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:

You may have an invalid version of smjavaagentapi.dll in the system path. Ensure that all installed products are compatible with one another and of compatible versions.

#### To check versions

1. Log in to the [Technical Support site](#).
2. Search for the SiteMinder Platform Support Matrix for r6.0 SP6

### Cookie Domain Mismatch Errors

#### Symptom:

After successful SAML authentication at consumer/SP site, user is still challenged by consumer/SP Web Agent because of cookie domain mismatch.

**Solution:**

Ensure that the producer/IdP and consumer/SP are not in the same cookie domain -- Federation Security Services does not support federation within the same cookie domain; it requires the use of separate cookie domains at the producer/IdP and consumer/SP sites. Additionally, you should ensure that the CookieDomainScope Web Agent parameter is set to the appropriate value for your environment (see information about single sign-on in the *SiteMinder Web Agent Guide*).

If separate cookie domains are in use, ensure that the cookie domain specified in the Agent configuration matches the domain name specified in the requested target URL.

## Error After Successful Authentication at Consumer/SP

**Symptom:**

After successful authentication at consumer site, an HTTP 404 "Page Not Found" error code is returned to user's browser.

**Solution:**

Ensure that the target page exists in the web server's document root. Check 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 consumer/SP site attempts to retrieve an assertion from the producer/IdP site, an HTTP 404 "Page Not Found" error code is returned to the user's browser.

**Solution:**

Ensure that the Federation Web Services application is deployed as a Web application on a web server running ServletExec or 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, as it fails to trust the Web server's certificate.

**Solution:**

Add the certificate of the Certificate Authority that issued the client certificate to the Web server's key database 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, for example, after "http:" in a web address, is case sensitive. Therefore, the case of the URLs in all corresponding settings must match.

For parameter values that should match between the Identity Provider/producer and Service Provider/consumer, refer to [Configuration Settings that Must Use the Same Values](#) (see page 503).

## 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:**

Ensure an affiliate domain has been created.

## Policy Server System Fails After Logoff

**Symptom:**

In some environments, when you log off from a system where the SiteMinder Policy Server service is running, the Policy Server service fails because of 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:**

Ensure that the anonymous user account associated with ServletExec has correct 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.

**More information:**

[Enable ServletExec to Write to the IIS File System](#) (see page 220)

## 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, located in the *web\_agent\_home*/WEB-INF/properties file
2. Add the following entry to the file just 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 SAML Affiliate Agent's default location.

#### To ensure 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.

## 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:**

In an environment in which 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, 2006 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, 2006 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:**

Ensure 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, ensure that the Name and Password values specified in the Affiliate Properties dialog at the producer site match the Affiliate Name and Password values configured for the SAML Artifact authentication scheme at the consumer site.
- If client certificate authentication is configured to protect the Assertion Retrieval Service, ensure that the consumer's client certificate is valid and that it is present in the consumer's AM.keystore database. Additionally, ensure 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 that protects the SAML 1.x Assertion Retrieval Service from Basic to Client Cert (or vice versa), subsequent authentication requests may fail.

**Solution:**

Restart the Web server after the authentication method is changed.

**More Information:**

[Access the Assertion Retrieval Service with a Client Certificate \(optional\)](#) (see page 276)

## 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 and gives following error 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 will fail.

**Solution:**

Ensure that a Web Agent response is created under the domain FederationWebServicesDomain. The response should 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, ensure 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, ensure that the Service Provider's client certificate is valid and that it is in the Service Provider's AM.keystore database. Additionally, ensure that the Certificate Authority that issued the client certificate is in the Web server's own key database at the Identity Provider.
- If no authentication is configured, ensure that the Artifact Resolution Service URL is *not* protected.

## ODBC Errors Deleting Expiry Data From Session Server

### Symptom:

On a Policy Server upgraded from an earlier version, ODBC errors occur when deleting expiry data from session server.

### Solution:

Upgrade the session server schema as described in the *SiteMinder Upgrade Guide*.



# Index

---

## A

- A Security Issue Regarding SAML 1.x Assertions • 251
- Access the Artifact Resolution Service with a Client Certificate (optional) • 390
- Access the Assertion Retrieval Service with a Client Certificate (optional) • 276
- Add a CA Certificate for an SSL Back Channel at the SP • 170
- Add a Client Certificate to smkeydatabase • 277
- Add a Consumer to an Affiliate Domain • 245
- Add a Domain Object • 240
- Add a Private Key and Certificate to the IdP Smkeydatabase • 174
- Add a Public Key to Smkeydatabase at the IdP • 177
- Add a Resource Partner to an Affiliate Domain • 407
- Add a SAML 2.0 Service Provider to an Affiliate Domain • 303
- Add Entities to an Affiliate Domain • 242
- Add Functionality to the Federation Deployment • 163
- Add the Service Provider to the Affiliate Domain at the IdP • 143
- Add the User Directory to the Affiliate Domain at the IdP • 143
- Add Users by Manual Entry for Access to a Service Provider • 305
- Add Users by Manual Entry for Resource Partner Access • 409
- Adding Users and Groups for Access to a Consumer • 248
- Adding Users by Manual Entry • 249
- Affiliate Attribute Editor Dialog • 517
- Affiliate Attribute Editor Dialog Fields and Controls • 517
- Affiliate Attribute Editor Dialog--Advanced Tab • 519
- Affiliate Attribute Editor Dialog--Attribute Setup Tab • 518
- Affiliate Attribute Editor Prerequisites • 517
- Affiliate Dialog • 509
- Affiliate Dialog Fields and Controls • 510
- Affiliate Dialog Prerequisites • 509
- Affiliate Dialog--Advanced Tab • 521
- Affiliate Dialog--Assertions Tab • 514
- Affiliate Dialog--Attributes Tab • 516
- Affiliate Dialog--IP Addresses Tab • 520
- Affiliate Dialog--Session Tab • 516
- Affiliate Dialog--Time Tab • 520
- Affiliate Dialog--Users Tab • 513
- Affiliate Domain Overview • 239
- Affiliation Overview • 349
- Affiliations for Single Logout • 350
- Affiliations for Single Sign-On • 349
- Affwebserver.log and FWSTrace.log Show Wrong Time • 645
- Aliases in the Smkeydatabase • 482
- Allow Access to the Federation Web Services Application • 256, 322, 420
- Allow Nested LDAP Groups Resource Partner Access • 409
- Allow Nested LDAP Groups Service Provider Access • 305
- Allow the Identity Provider to Assign a Value for the NameID • 319
- Allowing Nested Groups Access to Consumers • 249
- APIs for Federation Security Services • 83
- Artifact Resolution Service (SAML 2.0) • 626
- Assertion Consumer Service (SAML 2.0) • 636
- Assertion Consumer Services Dialog • 553
- Assertion Retrieval Service (SAML 1.x) • 625
- Assertion Validity for Single Sign-on • 252, 311, 413
- Assign a SAML 2.0 Authentication Scheme to a Realm • 379
- Assign an Administrator • 241
- Assign Name IDs to Affiliations • 350
- Assign the WS-Federation Scheme to a Realm • 451
- Assign User Directories • 241
- Attribute Authority and Query Reference • 589
- Attribute Service (SAML 2.0) • 629
- Attribute Svc Dialog • 589
- Attribute Svc Dialog Fields and Controls • 590
- Attribute Svc Dialog Prerequisites • 589
- Attribute Types • 259
- Attributes Dialog for Attribute Queries • 592
- Attributes Dialog Prerequisites • 591
- Attributes that Function for SSO and Attribute Query Requests • 330
- Auth. Scheme Properties--WS-Federation--Advanced Tab • 620

---

- Auth. Scheme Properties--WS-Federation--Signout Tab • 619
- Auth. Scheme Properties--WS-Federation--SSO Tab • 617
- Auth. Scheme Properties--WS-Federation--Users Tab • 616
- Authentication Fails After Modifying Authentication Method • 647
- Authentication Scheme Dialog -- SAML 2.0 Template • 565
- Authentication Scheme Dialog Prerequisites • 523
- Authentication Scheme Dialog--SAML 1.x Auth Scheme Properties (Artifact, POST) • 538
- Authentication Scheme Dialog--SAML Artifact Template • 524
- Authentication Scheme Dialog--SAML Artifact Template--Advanced Tab • 532
- Authentication Scheme Dialog--SAML Artifact Template--Scheme Setup Tab • 525
- Authentication Scheme Dialog--SAML POST Template • 532
- Authentication Scheme Dialog--SAML POST Template--Advanced Tab • 538
- Authentication Scheme Dialog--SAML POST Template--Scheme Setup Tab • 533
- Authentication Scheme Properties Dialog--WS-Federation Template • 611
- AuthnRequest (SAML 2.0) • 635
- AuthnRequest Query Parameters Used by a SiteMinder SP • 326

## B

- Benefits of SiteMinder Federation Security Services • 75
- Bindings for Single Logout • 377

## C

- CA Technologies Product References • iii
- Certificate Revocation Lists in the smkeydatabase • 484
- Certificates Stored in the SmkeyDatabase Only at the Consuming Authority • 484
- Choosing Whether to Protect the Intersite Transfer URL • 255
- Client Authentication Fails for SAML Artifact Single Sign-on • 647
- Command Options for smfedexport • 462
- Command Options for smfedimport • 468

- Configuration Checklist • 405
- Configuration Checklist at the Identity Provider • 301
- Configuration Checklist for 1.x Producer • 244
- Configuration Overview to Supply Attributes as HTTP Headers • 290, 372, 446
- Configuration Settings that Must Use the Same Values • 503
- Configuration Tasks for SAML 1.x Authentication • 276
- Configuration Tasks for SAML 2.0 Authentication • 356
- Configuration Tasks for WS-Federation Authentication • 433
- Configure a Custom Authentication Scheme for the Single Realm • 379
- Configure a Custom SAML 1.x POST Authentication Scheme • 283
- Configure a Custom WS-Federation Auth. Scheme • 452
- Configure a Default or Active Session Model • 257
- Configure a Name ID • 306
- Configure a Name ID for a WS-Federation Assertion • 410
- Configure a Name ID for Inclusion in the Assertion • 145
- Configure a Response to Send Attributes as HTTP Headers • 292, 374, 448
- Configure a SAML 2.0 Affiliation (Optional) • 307
- Configure a SAML or WS-Federation Authentication Scheme • 196
- Configure a Shared Session Model • 258
- Configure a Single Target Realm • 381, 453
- Configure a Single Target Realm for All SAML 1.x Auth. Schemes • 296
- Configure a Single Target Realm for All SAML Authentication Schemes • 379
- Configure a Single Target Realm for All WS-Federation Authentication Schemes • 452
- Configure a Single Use Policy • 367, 440
- Configure a Unique Realm for Each SAML 1.x Authentication Scheme • 296
- Configure a Unique Realm for Each SAML Authentication Scheme • 379
- Configure Affiliations • 350
- Configure an Affiliate Domain • 240
- Configure an Attribute Authority and a SAML Requester • 397
- Configure and Enable the Session Server • 208

---

Configure Assertion Attributes for WS-Federation • 417

Configure Attributes at the Attribute Authority • 398

Configure Attributes for Inclusion in Assertions (optional) • 329

Configure Attributes for SAML 1.x Assertions • 260

Configure Attributes for SSO Assertions • 331

Configure Attributes for WS-Federation Assertions (optional) • 417

Configure Attributes to Include in SAML 1.x Assertions (Optional) • 259

Configure Digital Signing (required for POST Binding) • 174

Configure Disambiguation Locally • 436

Configure Disambiguation Locally as Part of the Authentication Scheme • 362

Configure Federation Web Services (Consuming-side) • 198

Configure Federation Web Services (Producing-side) • 187

Configure FWS Trace Logging • 472

Configure IP Address Restrictions for 1.x Consumers (optional) • 262

Configure IP Address Restrictions for Service Providers (optional) • 320

Configure JBOSS or Tomcat to Work with Federation Web Services • 231

Configure POST Single Sign-on at the IdP • 146

Configure Request Processing with a Proxy Server • 338

Configure Request Processing with a Proxy Server at the SP • 376

Configure Required General Information • 308

Configure Required General Information for WS-Federation • 411

Configure SAML 1.x Artifact Authentication • 280

Configure SAML 1.x Assertions to Authenticate Users • 250

Configure SAML 1.x POST Profile Authentication • 282

Configure SAML 1.x User Disambiguation • 284

Configure SAML 2.0 Affiliations At the Identity Provider • 349

Configure SAML 2.0 Artifact Single Sign-on • 166

Configure SAML 2.0 SSO with Dynamic Account Linking at the SP • 70

Configure ServletExec to Work with Federation Web Services • 218

Configure Signout • 422

Configure Single Logout • 163, 378

Configure Single Logout (optional) • 333

Configure Single Sign-on at the SP • 364

Configure Single Sign-on for SAML 2.0 • 311

Configure Single Sign-on for WS-Federation • 412

Configure SiteMinder as a Resource Partner • 431

Configure SiteMinder as a SAML 1.x Consumer • 271

Configure SiteMinder as a SAML 1.x Producer • 243

Configure SiteMinder as a SAML 2.0 Identity Provider • 301

Configure SiteMinder as a SAML 2.0 Service Provider • 353

Configure SiteMinder as an Account Partner • 405

Configure SSO with Attributes from a Web Application • 67

Configure the AffWebServices.properties file • 155

Configure the AffWebServices.properties File • 214

Configure the AffWebServices.properties File at the IdP • 140

Configure the Back Channel for the Attribute Authority • 399

Configure the Backchannel for HTTP-Artifact SSO • 365

Configure the Backchannel for the Attribute Query • 402

Configure the Client Certificate Option at the Consumer • 277

Configure the Federation Web Services Properties Files • 213

Configure the Message Consumer Plug-in for WS-Federation • 442

Configure the NameID for the Attribute Query • 402

Configure the SAML 1.x AMAssertionGenerator.properties File • 203

Configure the SAML 1.x Artifact Scheme Setup • 280

Configure the SAML 1.x Message Consumer Plug-in • 287

Configure the SAML 2.0 Authentication Scheme • 359

Configure the SAML 2.0 Authentication Scheme at the SP • 157

Configure the Service Provider • 149

Configure the Web Server with the Web Agent Option Pack • 137, 153

Configure the WebLogic Reverse Proxy Plug-in • 225

Configure the WS-Federation Authentication Scheme • 434

Configure Time Restrictions for 1.x Consumers (optional) • 263

---

- Configure Time Restrictions for Service Provider Availability (optional) • 321
- Configure User Disambiguation for User Look Ups • 360
- Configure WebLogic to Work with Federation Web Services • 222
- Configure WebSphere to Work with Federation Web Services • 226
- Configure WS-Federation Single Sign-on at the Resource Partner • 438
- Configuring a SAML 1.x Assertion • 251
- Configuring the Client Certificate Option at the Service Provider • 390
- Configuring the Message Consumer Plug-in (SAML 2.0) • 386
- Considerations Before Migrating Key Databases • 500
- Consumer Not Authenticating When Accessing Assertion Retrieval Service • 646
- Contact CA Technologies • iii
- Conventions in the Installation Overview Procedures • 182
- Cookie Domain Mismatch Errors • 641
- Copy Web Agent Option Pack Libraries to WebSphere • 229
- Create a Custom SAML 2.0 Authentication Scheme (optional) • 360
- Create a Custom SAML Artifact Authentication Scheme (Optional) • 282
- Create a Custom WS-Federation Authentication Scheme • 435
- Create a Federation Attribute Variable • 403
- Create a Policy Expression with the Federation Attribute Variable • 404
- Create a Policy for the Target Resource • 384
- Create a Policy for the WS-Federation Resource • 455
- Create a Policy to Implement Attributes as HTTP Headers • 293, 375, 449
- Create a Policy to Protect the Authentication URL • 266
- Create a Response Associated with the Rule • 300
- Create a Rule for the Realm • 299
- Create a Rule for the Target Resource • 383
- Create a Rule for the Target WS-Federation Resource • 454
- Create a Unique Realm for Each WS-Federation Authentication Scheme • 451
- Create a WebAgent.conf File • 224, 229, 232

- Create an Authorization Rule to Validate Users • 291, 373, 447
- Create an SmHost.conf File • 224, 228, 232
- Create Links to Consumer Resources for Single Sign-on • 254
- Create Links to Initiate Single Sign-on (optional) • 200
- Create Links to Target Resources (optional) • 190
- Create the Artifact Resolution Service Policy • 346
- Create the Assertion Retrieval Service Policy • 268
- Create the SAML 1.x POST Common Setup and Scheme Setup • 282
- Creating Affiliate Domains • 239
- Customize a SAML Response Element (optional) • 340
- Customize Assertion Processing with the Message Consumer Plug-in • 286, 385, 441
- Customize SAML 1.x Assertion Content (optional) • 263
- Customize the Session Duration in the Assertion • 313
- Customizing Content in WS-Federation Assertions • 424
- Customizing SAML 2.0 Assertion Responses • 79

## D

- Debugging Features • 83
- Decrypt an Encrypted Assertion at the SP • 178
- Define Indexed Endpoints for Different Single Sign-on Bindings • 318
- Define Indexed Endpoints for the Assertion Consumer Service • 315
- Deploy a Federation Web Services WAR File in WebSphere • 230
- Deploy an FWS WAR File in JBoss or Tomcat • 233
- Deploy Federation Web Services as a Web Application • 217
- Deploy the FWS Application on WebLogic • 225
- Deploy the Sample Application on One System • 121
- Deploy the Sample Application on Two Systems • 122
- Deploying Federation with the FSS Sample Application • 113
- Deploying Federation without the FSS Sample Application • 127

---

## E

- Enable a Persistent Session to Store Assertions at the IdP • 167
- Enable Attribute Queries and Specify Attributes • 401
- Enable Encryption in the Policy Server User Interface at the IdP • 177
- Enable Identity Provider Discovery Profile (optional) • 335
- Enable Policy Server Trace Logging at the IdP • 136
- Enable ServletExec to Write to the IIS File System • 139, 220
- Enable Signature Validation at the SP • 176
- Enable Signout • 422, 441
- Enable Single Logout • 377
- Enable Single Logout at the IdP • 163
- Enable Single Logout at the SP • 164
- Enable SSL for the IdP Web Server for Artifact Single Sign-on • 167
- Enable the Artifact Binding for SAML Authentication at the SP • 171
- Enable the Creation of a Name Identifier • 320
- Enable the Enhanced Client or Proxy Profile • 369
- Enable Trace Logging for Federation Components at the SP • 152
- Enable Web Agent Option Pack Logging at the IdP • 141
- Enable Web Agent Option Pack Logging at the SP • 155
- Enabling Encryption • 336
- Encrypt a NameID and an Assertion • 336
- Encrypt and Decrypt the Assertion • 177
- Encrypted Private Key Fails to Be Imported into SMkeydatabase • 644
- Enforce a Single Use Policy to Enhance Security • 439
- Enforce Assertion Encryption Requirements for Single Sign-on • 384
- Enforce Policies that Protect Federation Web Services • 235
- Enforce the Authentication Scheme Protection Level for SSO • 319
- Enforcing a Single Use Policy to Enhance Security • 366
- Ensure the IIS Default Web Site Exists • 221
- Entities in a Federated Network • 72
- Environments that Require a Shared Session Store • 208

- Error After Successful Authentication at Consumer/SP • 642
- Error During Initialization of JVM • 645
- Error Message When Viewing FederationWSCustomUserStore • 643
- Exclude a User or Group from Service Provider Access • 304
- Excluding a User or Group from Access to a Consumer • 248
- Excluding a User or Group from Resource Partner Access • 408
- Export Metadata Tool • 458

## F

- Federated Single Sign-on with Security Zones • 74
- Federation Attribute Variable Editor Reference • 596
- Federation Authentication Scheme Dialog--Scheme Setup Tab • 613
- Federation Data Stored in the Session Server • 207
- Federation Sample Application Overview • 113
- Federation Security Services Concepts • 71
- Federation Security Services Overview • 23
- Federation Security Services Process Flow • 85
- Federation Security Services Trace Logging • 471
- Federation Services URLs • 623
- Federation Use Cases • 24
- Federation Web Services • 79
- Federation Web Services Access • 149
- Federation Web Services Application Overview • 211
- Federation Web Services Application Setup • 211
- Federation Web Services Deployment Descriptors • 213
- Federation Web Services Fails to Send SAML Request to Producer/IdP • 642
- Federation Web Services Template Sample • 476
- Federation Web Services URLs Used in SiteMinder Configuration • 623
- FederationSample.conf Settings • 118
- Flow Diagram for Authorizing a User with User Attributes • 396
- Flow Diagram for Identity Provider Discovery Profile • 108
- Flow Diagram for SAML 2.0 Single Logout • 101
- Flow Diagram for SSO Using SAML 1.x Artifact Authentication • 85
- Flow Diagram for SSO Using SAML 1.x POST Profile Authentication • 87

---

Flow Diagram for SSO Using SAML 2.0  
Authentication with Artifact Binding • 89  
Flow Diagram for SSO Using SAML 2.0  
Authentication with POST Binding • 93  
Flow Diagram for WS-Federation Signout  
(AP-initiated) • 103  
Flow Diagram for WS-Federation Signout  
(RP-initiated) • 106  
Flow Diagram for WS-Federation SSO Initiated at the  
Resource Partner • 97  
Flush Federation Web Services Cache for Trace Logs  
• 238  
Form the Policy to Protect the Target Resource • 300  
Formats Supported by the Smkeydatabase • 482

## G

General Issues • 641  
Guidelines for the Single Logout Confirmation Page •  
334

## H

How the Single Use Policy is Enforced • 367  
How the WS-Federation Single Use Policy is Enforced  
• 439  
How To Assign a SAML 1.x Authentication Scheme to  
a Realm • 295  
How To Protect a Resource with a SAML 1.x  
Authentication Scheme • 295  
How To Protect a Target Resource with a  
WS-Federation Authentication Scheme • 450  
How To Protect Resources with a SAML 2.0  
Authentication Scheme • 378  
How To Run the Sample Application • 118  
How to Use the Configuration Settings Tables • 503  
HTTP 404 Error When Trying to Retrieve Assertion at  
the Consumer • 642  
HTTP Error Handling at the IdP • 339

## I

Identify the SP, IdP, and Other General Settings •  
146  
Identity Provider Data for a Basic Configuration •  
129  
Identity Provider Data for an Advanced  
Configuration • 131  
Identity Provider Discovery Profile Service (SAML  
2.0) • 629  
Identity Provider Profiler Sample • 477

Identity Provider-initiated SSO (POST or artifact  
binding) • 323  
Implement WS-Federation Signout • 440  
Import Metadata Tool • 465  
Include an Allow/Create Attribute in Authentication  
Requests • 368  
Include an Attribute in the Assertion • 173  
Indexed Endpoints Flow Diagram • 316  
Initiate SAML 1.x Single Sign-On at the Producer •  
190  
Initiate SAML 2.0 Single Sign-On at the Identity  
Provider • 192  
Initiate SAML 2.0 Single Sign-on at the SP (optional) •  
200  
Initiate Single Sign-on at the Account Partner • 421  
Initiate Single Sign-on at the Resource Partner • 421  
Initiate WS-Federation Single Sign-on at the Account  
Partner • 192  
Initiate WS-Federation Single Sign-on at the  
Resource Partner • 201  
Install a Web Agent or SPS Federation Gateway  
(Consuming-side) • 197  
Install a Web Agent or SPS Federation Gateway  
(Producing-side) • 186  
Install a Web or Application Server for the Web  
Agent Option Pack (Consuming-side) • 197  
Install a Web or Application Server for the Web  
Agent Option Pack (Producing-side) • 186  
Install and Configure ServletExec to work with FWS  
at the IdP • 138  
Install and Configure ServletExec to Work with FWS  
at the SP • 154  
Install Federation Web Services at the Producer and  
Consumer • 279  
Install the Consuming-side Policy Server • 195  
Install the Consuming-side Policy Server Option Pack  
• 195  
Install the Consuming-side Web Agent Option Pack •  
198  
Install the IdP Policy Server • 134  
Install the IdP Policy Server Option Pack • 136  
Install the IdP Web Agent • 137  
Install the IdP Web Agent Option Pack • 137  
Install the JDK for Federation Web Services • 138,  
154  
Install the Policy Server and Its Option Pack • 279,  
357  
Install the Producing-side Policy Server • 184



---

Install the Producing-side Policy Server Option Pack • 184  
Install the Producing-side Web Agent Option Pack • 187  
Install the SP Policy Server • 150  
Install the SP Policy Server Option Pack • 152  
Install the SP Web Agent • 153  
Install the SP Web Agent Option Pack • 153  
Install the Web Agent or SPS Federation Gateway • 357  
Installation Overview • 181  
Integrate the Assertion Generator Plug-in with SiteMinder (SAML 1.x) • 264  
Integrate the Assertion Generator Plug-in with SiteMinder (SAML 2.0/WS-Federation) • 341, 425  
Integrate the Message Consumer Plug-in for SAML 1.x Authentication • 287  
Integrate the Message Consumer Plug-in with SiteMinder (SAML 2.0) • 387  
Integrate the Message Consumer Plug-in with SiteMinder (WS-Federation) • 443  
Internationalization in Federation Security Services • 84  
Intersite Transfer Service (SAML 1.x) • 624  
Introduction to SiteMinder Federation Security Services • 23

## J

Java Assertion Generator Plugin API • 84  
Java Message Consumer Plugin API • 84

## L

Locate User Records for Authentication • 436  
Log Messages for Federation Services at the Policy Server • 473  
Log Messages for Federation Web Services at the Web Agent • 471

## M

Manage the Key Database for Signing and Encryption • 479  
Manual Deployment Prerequisites • 128  
Manual FSS-to-FSS Deployment Overview • 127  
Matching Parameter Case-Sensitivity Configuration Issues • 643  
Migrate AM.keystore and Update smkeydatabase • 498

Modify the AffWebServices.properties File for JBOSS or Tomcat • 233  
Modify the AffWebServices.properties File for ServletExec • 219  
Modify the AffWebServices.properties File for WebLogic • 225  
Modify the AffWebServices.properties File for WebSphere • 229  
Modify the FederationSample.conf File • 118  
Modify the Key Database Using smkeytool • 486  
Multibyte Characters in Assertions are Not Handled Properly • 644

## N

Name IDs Dialog Fields for Attribute Queries • 594  
Navigate to the Affiliate Attribute Editor Dialog • 517  
Navigate to the Affiliates Dialog • 509  
Navigate to the Attribute Svc Dialog • 589  
Navigate to the Attributes Dialog • 592  
Navigate to the Auth. Scheme Properties Dialog--WS Federation Template • 612  
Navigate to the Authentication Scheme Dialog • 523  
Navigate to the Resource Partner Properties Dialog • 598  
Navigate to the Restrictions Dialog • 553  
Navigate to the SAML 2.0 Auth Scheme Properties Dialog • 566  
Navigate to the SAML Affiliation Dialog • 583  
Navigate to the SAML Service Provider Dialog • 541  
Navigating to the SAML Service Provider Attribute Editor Dialog • 555

## O

Obtain a LoginID for a WS-Federation User • 437  
Obtain the LoginID • 284, 362  
ODBC Errors Deleting Expiry Data From Session Server • 648  
Optional Configuration Tasks at a 1.x Producer • 245  
Optional Configuration Tasks for Configuring a Resource Partner • 406  
Optional Configuration Tasks for Identifying a Service Provider • 302  
Overview of a SiteMinder Federation Partnership Setup • 181

## P

Perform Authorizations with an Attribute Authority • 393

---

- Permit the Creation of a Name Identifier for SSO • 368
- Point the Policy Server to the IdP LDAP Policy Store • 134
- Point the Policy Server to the SP LDAP Policy Store • 150
- Policy Management API • 83
- Policy Server System Fails After Logoff • 643
- Prerequisites for Producing SAML 1.x Assertions • 243
- Prerequisites for Using the FSS Sample Application • 115
- Processing Import Files with Multiple Certificate Aliases • 469
- Processing Import Files with Multiple SAML 2.0 Providers • 469
- Properties File for the Key Database • 485
- Protect Federation Web Services (Consuming-side) • 199
- Protect Federation Web Services (Producing-side) • 188
- Protect Federation Web Services at the IdP (required-POST/Artifact) • 168
- Protect Target Resources (Consuming-side) • 196
- Protect the Artifact Resolution Service at the Identity Provider • 391
- Protect the Artifact Resolution Service with Client Certificate Authentication (optional) • 345
- Protect the Assertion Retrieval or Artifact Resolution Service (optional) • 236
- Protect the Assertion Retrieval Service at the Producer • 278
- Protect the Assertion Retrieval Service with Client Certificate Authentication (optional) • 267
- Protect the Authentication URL (SAML 2.0) • 147
- Protect the Authentication URL to Create a SiteMinder Session (SAML 1.x) • 265
- Protect the Authentication URL to Create a SiteMinder Session (SAML 2.0) • 343
- Protect the Authentication URL to Generate a SiteMinder Session • 427
- Protect the Federation Web Services Application • 234
- Protect the Target Resource at the SP • 158

## Q

- Query Parameter Processing by a SiteMinder IdP • 329

## R

- Request Processing with a Proxy Server at the IdP • 337
- Request Processing with a Proxy Server at the SP • 376
- Required Configuration Tasks at a 1.x Producer • 245
- Required Configuration Tasks for Configuring Resource Partners • 406
- Required Configuration Tasks to Identify a Service Provider • 302
- Resource Partner Attribute Dialog Prerequisites • 607
- Resource Partner Attribute Dialog--Advanced Tab • 609
- Resource Partner Attribute Dialog--Attribute Setup Tab • 607
- Resource Partner Properties Dialog • 597
- Resource Partner Properties Dialog Fields and Controls • 598
- Resource Partner Properties Dialog Prerequisites • 597
- Resource Partner Properties Dialog--Advanced Tab • 610
- Resource Partner Properties Dialog--Attributes Tab • 607
- Resource Partner Properties Dialog--General Tab • 603
- Resource Partner Properties Dialog--Name IDs Tab • 602
- Resource Partner Properties Dialog--Signout Tab • 609
- Resource Partner Properties Dialog--SSO Tab • 604
- Resource Partner Properties Dialog--Users Tab • 601
- Restrictions Dialog Field and Controls • 606
- Restrictions Dialog Fields and Controls • 554
- Restrictions Dialog for a Resource Partner • 606
- Restrictions Dialog for an SP • 553
- Restrictions Dialog Prerequisites • 554, 606
- Review Application-Generated SiteMinder Objects • 125
- Review the JVMOptions File Used to Create a JVM • 205
- Role of the Smkeydatabase at the Consuming Authority • 481
- Role of the Smkeydatabase at the Producing Authority • 481
- Run the migratekeystore Tool • 501
- Run the smfedexport Tool • 461



---

Run the smfedimport Tool • 466

## S

SAML 1.x Affiliate Dialog Reference • 509

SAML 1.x Artifact Authentication Scheme Overview • 274

SAML 1.x Artifact Profile Single Sign-On Failing • 646

SAML 1.x Assertion Generator Properties File • 203

SAML 1.x Authentication Scheme Prerequisites • 279

SAML 1.x Authentication Scheme Properties • 523

SAML 1.x Authentication Schemes • 271

SAML 1.x Matching Configuration Settings • 503

SAML 1.x POST Profile Authentication Scheme Overview • 273

SAML 1.x-Only Issues • 646

SAML 2.0 Attribute Authority Reference • 589

SAML 2.0 Attribute Query Reference • 591

SAML 2.0 Auth Scheme Properties Dialog • 568

SAML 2.0 Auth Scheme Properties Dialog Fields and Controls • 568

SAML 2.0 Auth Scheme Properties Dialog Prerequisites • 568

SAML 2.0 Auth Scheme Properties Dialog--Attributes Tab • 577

SAML 2.0 Auth Scheme Properties Dialog--Backchannel Tab • 578

SAML 2.0 Auth Scheme Properties Dialog--Encryption Tab • 577

SAML 2.0 Auth Scheme Properties Dialog--Name IDs Tab • 577

SAML 2.0 Auth Scheme Properties Dialog--SLO Tab • 575

**SAML 2.0 Auth Scheme Properties Dialog--SSO Tab • 571**

SAML 2.0 Auth Scheme Properties Dialog--Users Tab • 569

SAML 2.0 Auth. Scheme Properties Dialog--Advanced Tab • 579

SAML 2.0 Authentication Properties--Advanced Tab • 568

SAML 2.0 Authentication Properties--Scheme Setup Tab • 566

SAML 2.0 Authentication Scheme Overview • 353

SAML 2.0 Authentication Scheme Prerequisites • 357

SAML 2.0 Authentication Scheme Properties Reference • 565

SAML 2.0 Matching Configuration Settings • 505

SAML 2.0 Service Provider Reference • 541

SAML 2.0--Additional URL Configuration • 581

SAML 2.0--Additional URL Configuration Dialog • 562

SAML 2.0-Only Issues • 647

SAML Affiliate Agent • 82

SAML Affiliation Dialog Fields and Controls • 584

SAML Affiliation Dialog Prerequisites • 583

SAML Affiliation Dialog--Name IDs Tab • 585

SAML Affiliation Dialog--Users Tab • 586

SAML Affiliation Properties Dialog • 583

SAML Affiliations Reference • 583

SAML and WS-Federation Authentication Schemes • 79

SAML Assertion Generator • 77

SAML Auth Schemes Tab • 588

SAML Authentication Request Process • 355

SAML Browser Artifact Protocol • 80

SAML Credential Collector (SAML 1.x) • 634

SAML POST Profile Protocol • 80

SAML Profiles Supported by SiteMinder • 75

SAML Service Provider Attribute Editor Dialog Prerequisites • 556

SAML Service Provider Attribute Editor Dialog--Advanced Tab • 558

SAML Service Provider Attribute Editor Dialog--Attribute Setup Tab • 556

SAML Service Provider Dialog • 541

SAML Service Provider Dialog Prerequisites • 542

SAML Service Provider Dialog--Advanced Tab • 561

SAML Service Provider Dialog--Attributes Tab • 555

SAML Service Provider Dialog--Backchannel Properties • 549

SAML Service Provider Dialog--Encryption Tab • 560

SAML Service Provider Dialog--General Tab • 547

SAML Service Provider Dialog--IPD Tab • 559

SAML Service Provider Dialog--Name IDs Tab • 545

SAML Service Provider Dialog--SLO Tab • 558

SAML Service Provider Dialog--SSO Tab • 550

SAML Service Provider Dialog--Users Tab • 545

SAML Service Provider Properties Dialog Fields and Controls • 542

SAML Service Providers Tab • 588

Sample Federation Network • 128

Sample FSS Network • 117

Secure Proxy Server Federation Gateway • 82

Security Assertion Markup Language (SAML) • 71

Security Token Consumer Service (WS-Federation) • 637

Select the Artifact Binding at the IdP • 169

---

Select the Client Cert Option for Authentication • 277

Select Users for Which Assertions Will Be Generated • 247, 407

Select Users For Which Assertions Will Be Generated • 303

Select Users For Which Assertions Will Be Generated at the IdP • 144

Service Provider Data for a Basic Configuration • 132

Service Provider Data for an Advanced Configuration • 133

Service Provider Template Sample • 477

Service Provider-initiated SSO (POST or artifact binding) • 325

Set a Password for SAML Artifact Back Channel Authentication • 309

Set the Authentication Scheme Protection Level • 415

Set the Redirect Mode to Store SAML Attributes • 291, 373, 446

Set the ServletExec Library Path Variable • 219

Set the Sync Interval for Shared Sessions • 258

Set the WebLogic Library Path Variable • 223

Set the WebSphere Library Path Variable • 228

Set up a Key Database for Signing POST Responses • 190

Set Up a Key Database to Sign and Verify SAML POST Responses • 280, 358

Set up a SAML Requestor to Generate Attribute Queries • 400

Set up Affiliate Domains and Add Sites to these Domains • 185

Set up an Affiliate Domain at the IdP • 143

Set Up and Enabling Trace Logging • 471

Set Up Consuming Authority Components • 193

Set Up Encryption for SSO • 384

Set Up Links at the IdP or SP to Initiate Single Sign-on • 323

Set Up Links to Initiate WS-Federation Single Sign-on • 421

Set Up Producing Authority Components • 183

Set Up Sessions for a SAML Affiliate Agent Consumer (optional) • 257

Set Up smkeydatabase at the SP for Signature Validation • 175

Set up the Attribute Authority • 397

Set Up the Identity Provider • 133

Set Up the IdP Session Server for Artifact Single Sign-on • 166

Set Up the IdP User Store • 135

Set up the LoggerConfig.properties File • 215

Set Up the Service Provider • 149

Set Up the SP User Store • 151

Set up Time Restrictions for Resource Partner Availability (optional) • 416

Setup the SAML 1.x Assertion Generator File • 203

Set-up the smkeydatabase for Artifact Single Sign-on (optional) • 200

SetupFederationSample.pl Command Options • 119

Signout Service at the AP (WS-Federation) • 632

Signout Service at the RP (WS-Federation) • 639

Simplify Logging with Trace Configuration Templates • 475

Single Logout Request Validity • 334

Single Logout Service at the IdP (SAML 2.0) • 628

Single Logout Service at the SP (SAML 2.0) • 638

Single Sign On Service (SAML 2.0) • 627

Single Sign-on Service (WS-Federation) • 631

SiteMinder Components for Federation Security Services • 77

SiteMinder SAML 2.0 Metadata Tools Overview • 457

smfedexport Tool Examples • 464

smfedimport Tool Examples • 467

SmKeyDatabase Overview • 479

Smkeytool Command Options • 489

Smkeytool Command Syntax and Options • 487

Smkeytool Examples for UNIX Platforms • 495

Smkeytool Examples for Windows Platforms • 497

Solution 1

    Single Sign-on based on Account Linking • 37

Solution 1 Using SAML 1.x Artifact Authentication • 38

Solution 1 Using SAML 1.x POST Profile • 39

Solution 1 Using SAML 2.0 Artifact Authentication • 40

Solution 1 Using SAML 2.0 POST Binding • 41

Solution 1 Using WS-Federation Passive Requestor Profile • 43

Solution 10

    Single Sign-on with No User ID at the IdP • 59

Solution 11

    SAML Artifact SSO Using Security Zones • 61

Solution 12

    SAML 2.0 SSO with Attributes from a Web Application • 64

Solution 13

---

- SAML 2.0 SSO with Dynamic Account Linking at the SP • 68
- Solution 2
  - Single Sign-on based on User Attribute Profiles • 44
- Solution 3
  - Single Sign-on with no Local User Account • 46
- Solution 4
  - Extended Networks • 49
- Solution 5
  - Single Logout (SAML 2.0) • 50
- Solution 6
  - WS-Federation Signout • 52
- Solution 7
  - Identity Provider Discovery Profile (SAML 2.0) • 54
- Solution 8
  - Multi-protocol Network • 56
- Solution 9
  - SAML 2.0 User Authorization Based on a User Attribute • 58
- Solutions for Federation Use Cases • 36
- SP Not Authenticating When Accessing Assertion Retrieval Service • 648
- Specify a Search Specification to Locate a User (SAML 1.x) • 285
- Specify IP Address Restrictions for Resource Partners (optional) • 415
- Specify Name Identifiers for SAML 2.0 Assertions • 306
- Specify Name IDs for WS-Federation Assertions • 410
- Specify Redirect URLs for Failed SAML 1.x Authentication • 293
- Specify Redirect URLs for Failed SAML 2.0 Authentication • 388
- Specify Redirect URLs for Failed WS-Federation Authentication • 449
- Specify the POST Binding Authentication at the SP • 156
- Specify the User Store for the IdP Policy Server • 142
- Specify Users for Disambiguation for SAML Affiliations • 351
- Storing User Session, Assertion, and Expiry Data • 207
- Supply SAML Attributes as HTTP Headers • 288, 370, 444

## T

- Test Federation Web Services • 156
- Test Federation Web Services at the IdP • 141
- Test SAML 2.0 Single Sign-on • 160
- Test Single Logout • 165
- Test Single Logout with the FSS Sample Application • 124
- Test Single Sign-on with the FSS Sample Application • 123
- Test SP-Initiated Artifact Single Sign-on • 172
- The JVMOptions.txt File • 205
- The Web.xml File • 640
- Trace Logging • 471
- Trace Logging Templates for Federation Web Services • 475
- Trace Logging Templates for the IdP and SP • 476
- Trace Logs Not Appearing for IIS Web Server Using ServletExec • 644
- Troubleshooting • 641

## U

- Unsolicited Response Query Parameters Used by a SiteMinder IdP • 324
- Update Federation Web Services Data in the Logs • 474
- URLs for Services at the Consuming Authority • 633
- URLs for Services at the Producing Authority • 623
- Use a Client Cert. to Protect the Assertion Retrieval or Artifact Resolution Service • 237
- Use a SAML Affiliation to Locate a User Record (Optional) • 361
- Use a Script to Create a New Attribute • 419
- Use a Script to Create A New Response Attribute • 262
- Use a Search Specification to Locate a User • 363
- Use a Search Specification to Locate a WS-Federation User • 437
- Use an Attribute Authority to Authorize Users • 393
- Use Basic over SSL Scheme to Protect the Assertion Retrieval Service • 236
- Use Case 1
  - Single Sign-on Based on Account Linking • 24
- Use Case 10
  - SAML 2.0 Single Sign-on with No Name ID at the IdP • 32
- Use Case 11
  - SAML Artifact SSO Using Security Zones • 33
- Use Case 12

---

- SAML 2.0 SSO Using Attributes from a Web Application • 34
- Use Case 13
  - SSO with Dynamic Account Linking at the SP • 36
- Use Case 2
  - Single Sign-on Based on User Attribute Profiles • 25
- Use Case 3
  - Single Sign-on with No Local User Account • 26
- Use Case 4
  - Extended Networks • 27
- Use Case 5
  - Single Logout • 28
- Use Case 6
  - WS-Federation Signout • 29
- Use Case 7
  - Identity Provider Discovery Profile • 30
- Use Case 8
  - Multi-protocol Support • 31
- Use Case 9
  - SAML 2.0 User Authorization Based on a User Attribute • 31
- Use Case for SAML Attributes As HTTP Headers • 289, 371, 444
- Use of Client Cert. Auth. with an IIS 5.0 Web Server • 268
- Use SAML 2.0 Provider Metadata To Simplify Configuration • 457
- Use the SiteMinder Profiler to Log Trace Messages • 474
- User Mapping • 73
- Using a Script to Create A New Attribute • 332

## V

- Validate Signed AuthnRequests and SLO Requests/Responses • 309
- Validate Signout Requests that are Digitally Signed • 423
- View a List of Service Providers in an Affiliation • 352
- View Authentication Schemes That Use an Affiliation • 352

## W

- Web Agent Option Pack Fails to Initialize Due to Invalid smjavaagent.dll • 641
- WebLogic Configuration Required for Back Channel Authentication • 366
- What Gets Stored in smkeydatabase? • 483

- Where to Find Federation Security Services Information • 111
- WSFedDispatcher Service at the AP • 633
- WSFedDispatcher Service at the RP • 640
- WS-Federation • 72
- WS-Federation Assertion Generator • 78
- WS-Federation Auth. Scheme Fields and Controls • 615
- WS-Federation Auth. Scheme Properties Dialog Prerequisites • 612
- WS-Federation Authentication Scheme Dialog--Advanced Tab • 614
- WS-Federation Authentication Scheme Overview • 431
- WS-Federation Authentication Scheme Prerequisites • 433
- WS-Federation Authentication Scheme Reference • 611
- WS-Federation Configuration Settings • 506
- WS-Federation Passive Requestor Profile Protocol • 81
- WS-Federation Resource Partner Properties Reference • 597
- WS-Federation SSO Initiated at the Account Partner • 101